

**NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA**



THESIS

**APPLICATION OF MULTI-BLOCK CFD
TECHNIQUES TO A MISSILE GEOMETRY**

by

Bret S. Barton

December, 1995

Thesis Advisor:

Garth Hobson

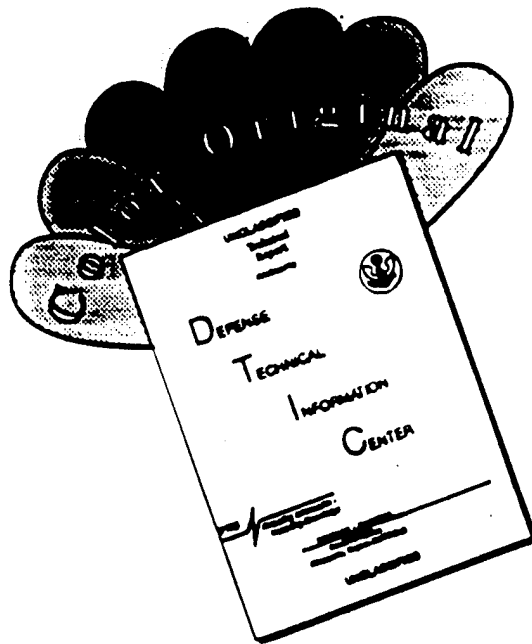
Approved for public release; distribution is unlimited.

19960507 081

DTIC QUALITY INSPECTED 1

19960507 179

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF COLOR PAGES WHICH DO NOT REPRODUCE LEGIBLY ON BLACK AND WHITE MICROFICHE.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 1995		3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE APPLICATION OF MULTI-BLOCK CFD TECHNIQUES TO A MISSILE GEOMETRY			5. FUNDING NUMBERS	
6. AUTHOR(S) Bret S. Barton				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) The aerodynamics of a missile body were modeled using computational fluid dynamics (CFD) techniques. A multi-block approach was used on a slender body and intersecting symmetric thin delta-wing. The CFD process and software were examined thoroughly including multi-block grid generation and interpolation, iblanking methods and flow-solver analysis. CFD results were compared with available wind tunnel data. Two Cartesian free-stream grids, a wing C-grid, a collar and body grid were used to model the body/wing geometry. The wing grid had a sharp tip and sharp leading and trailing edges. The body/wing intersection was represented with the collar grid. Both a hyperbolic grid generator, HYPGEN and an elliptic grid generator, GRIDGEN Vr 9, were evaluated. PEGSUS Vr 4.0 was used to compute the iblanking and interpolation stencil, based on the Chimera overlapping grid scheme. A single composite mesh was passed to the Navier-Stokes implicit flow-solver OVERFLOW Vr 1.6ag. Solutions were computed for inviscid and viscous flows at different Mach numbers and incidence angles. The Baldwin-Lomax shear and boundary layer turbulent models were used. Agreement was found between published wind tunnel data and the CFD solution thus validating the grid generation and flowfield solution procedure.				
14. SUBJECT TERMS multi-block, computational fluid dynamics, grid generation			15. NUMBER OF PAGES 105	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

Approved for public release; distribution is unlimited.

**APPLICATION OF MULTI-BLOCK CFD TECHNIQUES TO A MISSILE
GEOMETRY**

Bret S. Barton
Captain, Canadian Armed Forces
B.Eng., Technical University of Nova Scotia, 1989

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

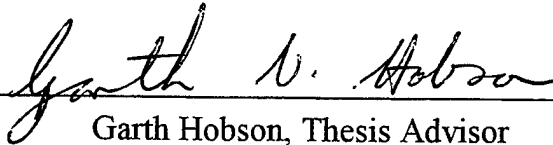
December 1995

Author:

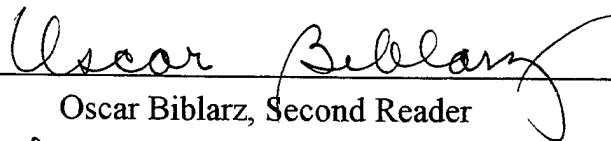


Bret S. Barton

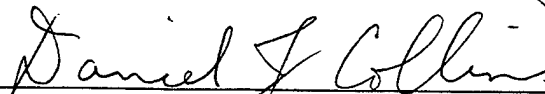
Approved by:



Garth Hobson, Thesis Advisor



Oscar Biblarz, Second Reader



Daniel J. Collins, Chairman

Department of Aeronautics and Astronautics

ABSTRACT

The aerodynamics of a missile body were modeled using computational fluid dynamics (CFD) techniques. A multi-block approach was used on a slender body and intersecting symmetric thin delta-wing. The CFD process and software were examined thoroughly including multi-block grid generation and interpolation, iblanking methods and flow-solver analysis. CFD results were compared with available wind tunnel data. Two Cartesian free-stream grids, a wing C-grid, a collar and body grid were used to model the body/wing geometry. The wing grid had a sharp tip and sharp leading and trailing edges. The body/wing intersection was represented with the collar grid. Both a hyperbolic grid generator, HYPGEN and an elliptic grid generator, GRIDGEN Vr 9, were evaluated. PEGSUS Vr 4.0 was used to compute the iblanking and interpolation stencil, based on the Chimera overlapping grid scheme. A single composite mesh was passed to the Navier-Stokes implicit flow-solver OVERFLOW Vr 1.6ag. Solutions were computed for inviscid and viscous flows at different Mach numbers and incidence angles. The Baldwin-Lomax shear and boundary layer turbulent models were used. Agreement was found between published wind tunnel data and the CFD solution thus validating the grid generation and flowfield solution procedure.

TABLE OF CONTENTS

I. INTRODUCTION	1
II. MULTI-GRID GENERATION	3
A. OVERVIEW	3
B. GRIDGEN SOFTWARE DESCRIPTION	5
1. GRIDGEN Overview	5
2. Grid Generation Process	5
a. Connector Description	6
b. Domain Description	7
c. Block Description	8
C. HYPGEN SOFTWARE DESCRIPTION	8
D. PEGSUS SOFTWARE DESCRIPTION	9
E. OVERFLOW SOFTWARE DESCRIPTION	10
III. SLENDER-BODY ANALYSIS	11
A. OVERVIEW	11
B. GEOMETRY DEFINITION	11
C. GRID GENERATION	13
1. Body Grid	13
2. Outer Cartesian Grid	15

3.	Symmetry Plane Addition	17
D.	PEGSUS IBLANK PROCEDURE	17
1.	Boundary Definitions	17
a.	Outer Boundary	18
b.	Hole Boundary	19
2.	Orphan Point Elimination	20
a.	Interpolation Stencil Requirements	20
b.	Elimination Techniques	21
E.	OVERFLOW ANALYSIS	22
1.	Input Conditions	22
2.	Results	23
IV.	BODY AND WING ANALYSIS	27
A.	OVERVIEW	27
B.	GRID GENERATION	28
1.	Body Grid	29
2.	Wing Grid	29
3.	Collar Grid	30
a.	Wing/Body Intersecting Line	31
b.	Body Conforming Collar Surface	31
c.	Wing Conforming Collar Surface	33
d.	Surface Grid Concatenation	35

e.	Volume Grid Generation	35
4.	Cartesian Grids	36
C.	PEGSUS IBLANK PROCEDURE	37
D.	OVERFLOW ANALYSIS	37
1.	Boundary Conditions	38
2.	Turbulence Modeling	38
a.	Baldwin-Lomax Boundary Layer Model	38
b.	Baldwin-Lomax Shear Layer Model	39
3.	Results	39
a.	Zero Degrees AOA	41
b.	8.7 Degrees AOA	43
c.	19.3 Degrees AOA	49
V.	CONCLUSIONS AND RECOMMENDATIONS	55
	LIST OF REFERENCES	59
	APPENDIX A. PGSUS INPUT FILES	61
	APPENDIX B. OVERFLOW INPUT FILES	67
	APPENDIX C. MATLAB SCRIPT FILES	71
	APPENDIX D. COLLAR AND HYPGEN INPUT FILES	77
	APPENDIX E. L_2 NORM, C_L & C_M PLOTS	83
	INITIAL DISTRIBUTION LIST	93

I. INTRODUCTION

The advent of supercomputers has made possible the numerical solution of the Navier-Stokes equations applied to complex flows. Computational fluid dynamics (CFD) has been used in many aeronautical configurations of which there are numerous packages available. Many grid generation packages are also available for geometry and flow-field definition.

Creating a computational grid that accurately represents the object of study constitutes most of the effort involved in a CFD analysis, once a suitable flow solver has been developed. The grid generation procedure involves defining the solid geometry of the structure to be modeled, creating surface meshes that represent the object and finally incorporating these meshes into a grid block structure that encloses the object and surrounding free-stream space.

Creating a single-block grid around a complex body while maintaining the required grid density and orthogonality is difficult. Complex structures are often modeled by defining several single-block grids that overlap and together define the entire computational space [Ref. 1]. Communication between multiple over-set grids requires an established protocol. The Chimera over-set grid scheme controls the overlapping requirements of multiple blocks [Refs. 2 and 3]. The multi-block grid network is then merged to form a single composite-grid that can be used in a single-ordered grid flow solver.

NASA Ames Research Center (ARC) currently uses and supports a CFD code (OVERFLOW) which has extensively modelled the Space Shuttle vehicle aerodynamics [Ref. 4]. The Naval Postgraduate School (NPS) has successfully applied OVERFLOW to single-block grid geometries at various flight conditions [Ref. 5]. NASA ARC also applied the multi-block analysis to missile configurations [Ref. 6]. The first attempt at NPS with multi-block analysis was made by Reuter [Ref. 7]. He contributed to the multi-grid analysis of an integrated Space Shuttle canard; however, the efforts were centralized at NASA ARC and specific grid generation methodologies were not recorded.

NPS has the resources to facilitate significant study in aeronautical CFD methods. The major obstacle has been a lack of clear understanding of the research software capabilities, limitations and input requirements. The objective of this thesis was to clearly define the multi-grid CFD process. To this end, software was evaluated and procedures recorded to effectively streamline future efforts.

Mesh generators typically solve either elliptic or hyperbolic partial differential equations which determine grid distributions. Both methods have their strengths when attempting to build grids around extreme gradients such as those defining a supersonic thin airfoil. A successful grid procedure was developed to model the flow around a thin, symmetric delta-wing intersecting a missile shaped body. The grid requirements around sharp leading and trailing edges, sharp wing tips and extreme concave wing/body intersections were also investigated.

II. MULTI-GRID GENERATION

A. OVERVIEW

The multi-grid approach can be accomplished by defining each grid such that they have a common boundary of abutting blocks or an overlapping grid structure. The abutting grids need not necessarily have point-to-point correspondence. The overlapping method is called the Chimera scheme and is supported by the flow solver OVERFLOW [Ref. 4]. The Chimera scheme will be referred to in the rest of this thesis.

The grid generation software used to create the multi-grid representation wing/body model was GRIDGEN Version 9 [Ref. 8] and HYPGEN Version 1.3 [Ref. 9]. The Chimera overlapping scheme handles grid generation around complex aerodynamic shapes. One large intricate grid is replaced by several simpler grids, each representing components of the complete configuration. Communication between meshes is coordinated by PEGSUS [Ref. 10] which produces an interpolation file used by OVERFLOW.

A CFD analysis is an iterative process as depicted by the flowchart in Figure 1. Ingenuity of grid design is often required when modelling complex geometries. Once the grids have been developed, either by HYPGEN or GRIDGEN, PEGSUS determines the interpolation stencil based on an input file. A successful interpolation map and input file, describing initial boundary conditions, are used by the flow solver. Desired output from PEGSUS and OVERFLOW are achieved by successive grid refinement and accurate input.

GRIDGEN - MDA Eng.
(NASA ARC sponsored)

HYPGEN - NASA ARC

NASA ARC

Arnold Eng. Dev. Ctr.

Object to be Modeled
- CAD

Note 1 - GRIDGEN O/P is ASCII formatted
with grid number on the first line. Remove
this line and use GRIDED to change format
to unformatted

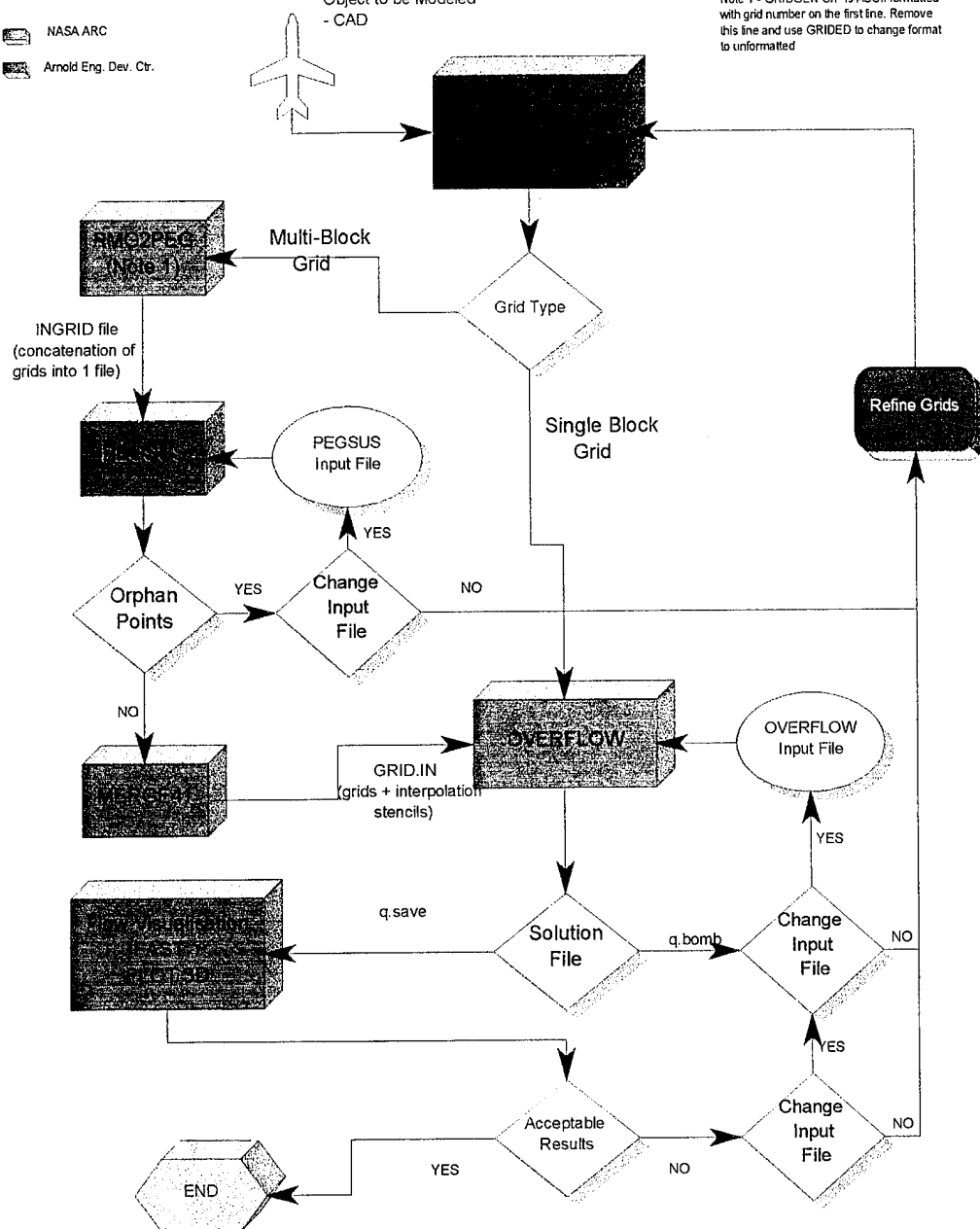


Figure 1. CFD Procedure Flow Chart

B. GRIDGEN SOFTWARE DESCRIPTION

1. GRIDGEN Overview

GRIDGEN is an interactive code used to generate three-dimensional grids around bodies, within user defined blocks. It can distribute grid points on curves, initialize and refine grid points on surfaces and initialize volume grid points. GRIDGEN Version 9, sponsored by NASA Ames Research Center and developed by Computer Sciences Corporation, was written using the Silicon Graphics Iris GL graphics library and runs on Silicon Graphics 4D Series and IBM RS/6000 Series workstations. [Ref. 8]

GRIDGEN is not a computer aided design (CAD) package and as such does not have the tools to define complex geometries but can generate simple three-dimensional and most two-dimensional shapes. The first step in grid generation is to either draw the object in a CAD package and import it into GRIDGEN or generate the required shapes directly. The only purpose of the CAD surface generation is to define the object and this usually has no relationship to the grid topology or quantity of grid points.

2. Grid Generation Process

Creating a volume grid in GRIDGEN requires following a set of successive steps that include:

1. defining the outer boundaries of the grid by creating a series of continuous segments, called connectors, which have grid points defined and distributed along their length,

2. generating a four edged mesh called a domain, which is smoothed using algebraic or elliptic smoothing routines, and
3. grouping the domains together to form a viable computational block and smoothing the final three-dimensional volume grid.

Figure 2 shows the relationship between the physical grid block generated in GRIDGEN and its equivalent computational block.

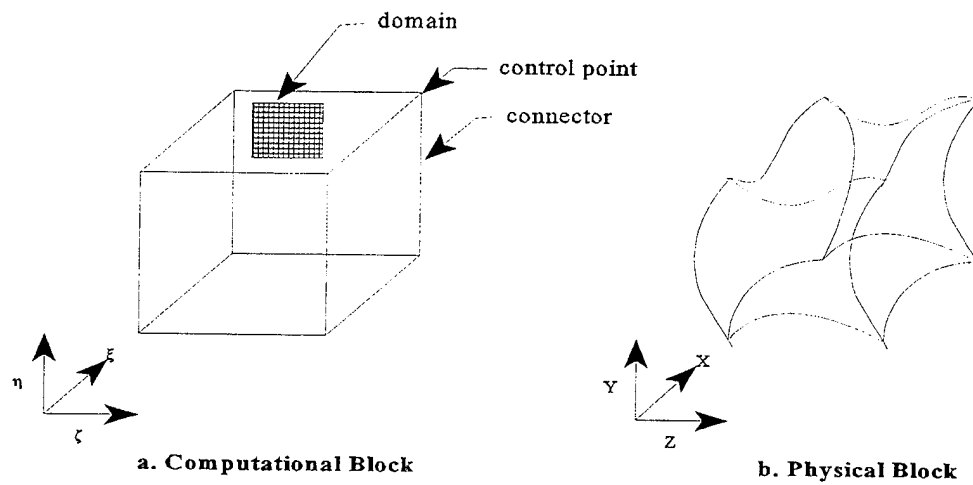


Figure 2. Relationship Between Computational and Physical Blocks

a. Connector Description

Connectors consist of line and curve segments that form the outer boundary structure of the grid. Each connector begins and ends with a control point and may consist of several sub-connectors.

Segments within a connector are dimensioned and grid points are distributed along the segment. Grid point locations are controlled by geometric or linear distribution functions. Specific controls are available to dictate exact grid point spacing parameters.

b. Domain Description

A domain is a surface mesh defined by four edges. Each edge consists of one or more connectors joined through control points. The quantity of grid points on each edge forming a domain must match its opposite edge. There are no limits on the size of the domains; however, the domain shape will affect the contours of the final volume grid.

Smoothing the domains provides a method of modifying interior grid point locations. GRIDGEN automatically uses an algebraic transfinite interpolation (TFI) method of smoothing the generated domains. Several algebraic smoothing routines were available which were best applied to Cartesian type domains or domains shaped similarly to a body of revolution. Domains which were created and confined to the shape of a CAD database could also be smoothed algebraically. An elliptic smoother was also available particularly for domains with areas of high gradients. Effective smoothing helped ensure a gradual transition of grid spacing and varying aspect ratio in high curvature areas such as wing leading edges, or missile nose sections. Control over the shape at the boundaries was effected by applying edge constraints.

c. Block Description

Blocks were defined by six faces; each composed of one or more domains. The face provided sufficient boundary conditions to initialize interior grid points. Each block was its own individual volume grid and when combined formed a multi-block grid.

Blocks were smoothed using the default algebraic TFI method, however these could have been smoothed elliptically using GRIDGEN3D, a software program external to GRIDGEN. GRIDGEN would produce an initial estimate of the required input file for GRIDGEN3D; however, specific boundary conditions constraining the elliptic smoothing routines had to be entered.

C. HYPGEN SOFTWARE DESCRIPTION

HYPGEN was a NASA ARC code used to grow a three-dimensional grid from a user-supplied surface grid. HYPGEN solved the three-dimensional hyperbolic equations, which were two orthogonality relations and one cell volume constraint. [Ref. 9]

GRIDGEN was used to produce the surface grid defining the object. The volume grid was then grown by marching away from the body (in the L direction in a J,K,L grid indexing system) according to the boundary conditions specified in the user input file. The user controlled parameters included the initial spacing away from the surface, stretching constraints, J and K boundary conditions including symmetry and periodicity and smoothing controls.

HYPGEN performed grid quality checks every time a grid was generated. The first and most stringent test was the cell volume computation by tetrahedron decomposition. The second was a Jacobian computation using the same algorithm as in OVERFLOW. Therefore if a grid passed the Jacobian test it would pass the initial test in the flow solver; if it failed the first test the grid was probably distorted and accuracy would be degraded. [Ref. 9]

D. PEGSUS SOFTWARE DESCRIPTION

PEGSUS, developed by the Arnold Engineering Development Center (AEDC), facilitated the multi-grid approach to CFD modeling. It produced a composite mesh file, which was a concatenation of all the grids. Also generated was an interpolation file, which associated all interpolated boundary points in the composite mesh with mesh points that supplied the interpolated flow-field values.

Modeling a complex geometry with multiple grids would yield varying degrees of overlap. Redundant grid points were identified as either hole points or interpolation boundary points and the remainder were field points. PEGSUS identified and labeled all the grid points in an IBLANK file that were passed to the flow solver. Hole and interpolation points (IBLANK = 0) were excluded from the computational domain or updated by interpolation. Field points (IBLANK = 1) were updated by the solution algorithm and associated boundary conditions.

PEGSUS provided several methods of identifying hole points within a grid. Defining surfaces within a grid topology identify the hole boundaries. Grid points enclosed by the outer boundary of the accompanying mesh and the hole boundary formed the interpolation region. When insufficient points were available to form a satisfactory interpolation stencil, the grid point was labeled an orphan point. Orphan points were eliminated by ensuring adequate grid overlap at the boundaries.

E. OVERFLOW SOFTWARE DESCRIPTION

OVERFLOW is an implicit flow-solver which was written and developed at NASA ARC. The code solved the Reynolds-Averaged Navier Stokes equations in strong conservative form, and used the grid blanking information from PEGSUS and the initial boundary conditions to compute the flow solution. User controlled parameters included:

1. basic flow properties such as the angle-of-attack, sideslip angle, Reynolds number, free-stream Mach number;
2. variations in the properties of the ratios of specific heats (γ) ;
3. solution controls such as time stepping, stability parameters, differencing schemes and smoothing;
4. boundary conditions applied to symmetry planes, outer grid boundaries, solid surfaces, C-grid "cuts"; and
5. turbulence model types which include Baldwin-Lomax boundary and shear layer models and the Baldwin-Barth one equation (k) model.

III. SLENDER-BODY ANALYSIS

A. OVERVIEW

Completing the flow analysis around a simple slender-body provided an efficient means of learning the intricacies of grid generation and flow solution in a multi-block grid system. A number of goals were established prior to commencement. They were :

1. validate and become efficient with the grid generation routines;
2. determine the interactions between software programs investigating compatibility issues of file types and platforms;
3. demonstrate a valid two-grid Chimera solution using PEGSUS and OVERFLOW;

PEGSUS and OVERFLOW code installations on the CRAY Y-MP EL98 were validated. Supplied with the OVERFLOW code were HYPGEN examples of wing/body multi-block grids, PEGSUS input files and OVERFLOW input files. Flow solutions were successfully computed with the examples provided. The procedure identified problems in the limits placed on dimensioned variables set on compilation, differences between software version requirements and IRIS UNIX and CRAY UNIX file compatibility. Dimension limits were reset, software recompiled and software version compatibility established.

B. GEOMETRY DEFINITION

Definition of the slender-body was done using AutoCAD though the shape was simple enough to be generated directly in GRIDGEN. The dimensions for the body were taken from

the experimental investigation [Ref. 11] . The IGES format AutoCAD file was imported into GRIDGEN as a database entity. Additional information on acceptable database types are located in chapter 3.3.6 of the GRIDGEN user manual [Ref. 8].

The slender-body was symmetric in the Y-plane therefore only half the body was modeled. The fore-body and aft-body were fit to the following polynomials $0.5*(1-\{(x-2)/2\}^{**4})$ and $0.5*(1-\{(x-8.05)/2\}^{**4})$ respectively. The body was 10.05 units in length and 1 unit wide. Figure 3 details the relative dimensions of the slender-body and wing (modeled in the follow-on analysis).

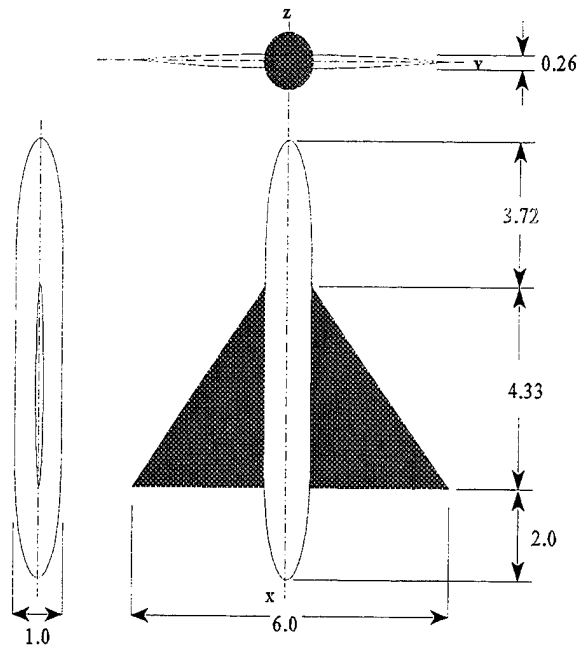


Figure 3. Solid Body Dimensions

C. GRID GENERATION

Two overlapping grids were required to map the flow field around the body. A dense inner grid enclosing the slender-body defined the viscous surface, the boundary layer and surrounding flow extending three body diameters from the surface. The free-stream flow beyond the body grid was represented by a coarse Cartesian grid.

1. Body Grid

Grid generation around the body included defining the surface grid, building the volume grid framework with connectors, defining the domains and compiling the block with GRIDGEN. The surface grid was generated using an algebraic-polar smoothing routine which treated the geometry as a body of revolution. The blue connector lines, displayed in Figure 4, formed the volume grid framework and included two stings along the X-axis ahead of and behind the slender body. The right-handed grid ($51 \times 59 \times 35$) extended three body diameters in all directions. Grid lines with increasing J indices ran the length of the body, grid lines with increasing K indices ran around the body from bottom to top and grid lines normal to the surface had increasing L indices. Grid points were clustered about the nose and tail. Grid cells with a unit aspect ratio were desired in high gradient regions. Initial node grid spacing along the body of 0.0025 was increased geometrically to 0.5 through the mid section. Grid aspect ratio growth was limited to twenty percent increments.

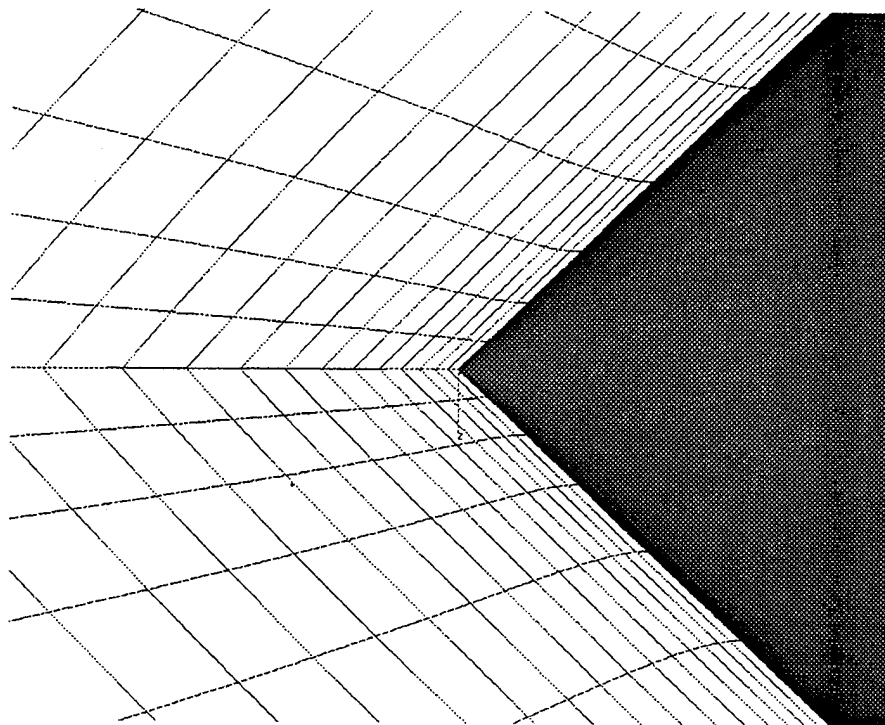


Figure 4. Fore-body Grid Spacing

The grid was generated away from the body with an initial spacing of 0.0006 and then smoothed elliptically. Figure 5 shows the grid spacing around the fore body. Clustering of grid planes within the boundary layer while maintaining orthogonality along the solid boundaries were requisite for accurate viscous solutions.

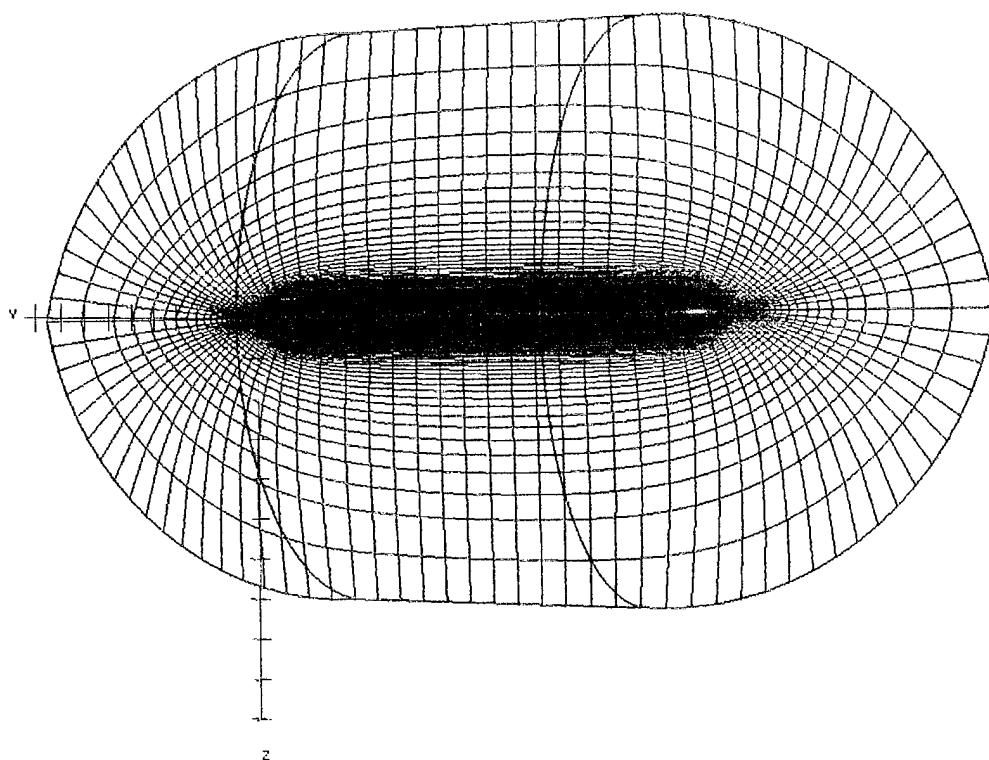


Figure 5. Slender Body Grid Scheme

2. Outer Cartesian Grid

A $50 \times 30 \times 21$ Cartesian grid, which extended three body lengths in all directions, enclosed the slender-body grid. The grids were concentric in the XZ plane with a common Y - symmetry plane ($L = 1$). Regions of Cartesian grid clustering were centered about the

outer plane of the body grid ($L = 35$) to ensure sufficient overlap in the interpolation region. The convex body-grid shape created overlapping difficulties requiring several iterations to match the two grids. Grid stretching ensured a smooth aspect ratio transition between grids. Figure 6 displays the overlapping structure of the grids with the J indexing from left to right, K lines ran from bottom to top and L lines away from the Y-symmetry plane. The irregular cut-out regions in the Cartesian grid were a result of the PEGSUS blanking.

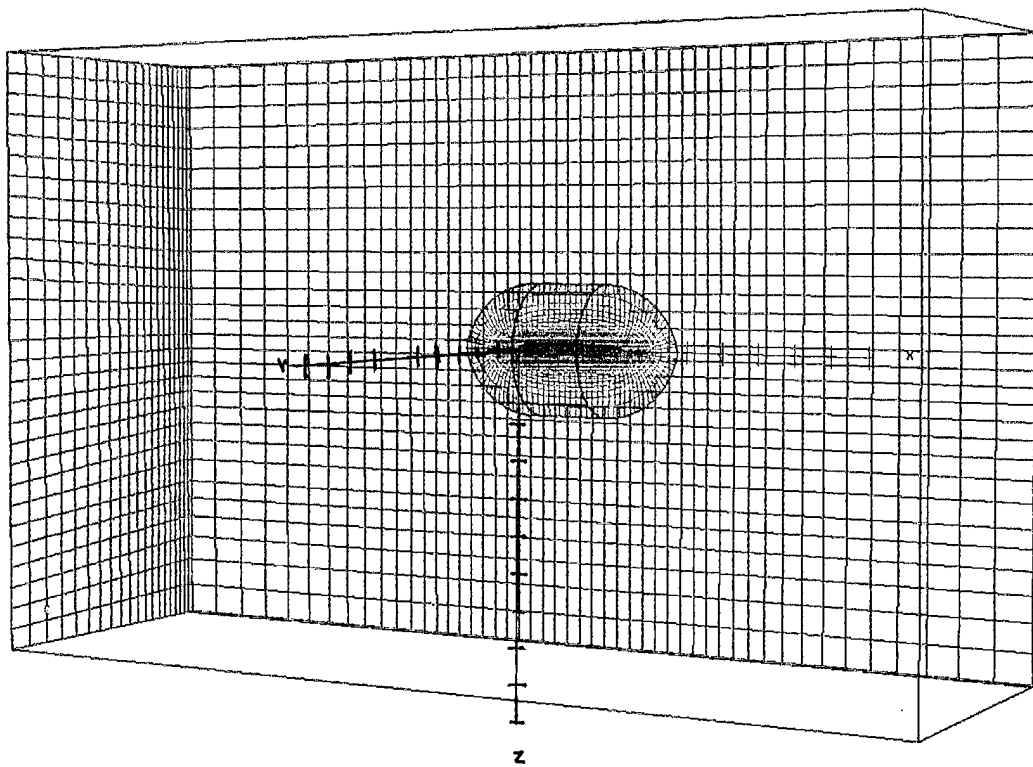


Figure 6. Cartesian and Slender Body Overlapping Grid Structure

3. Symmetry Plane Addition

The OVERFLOW solver required the first plane beyond the symmetry plane be provided when utilizing the symmetry boundary condition. The slender-body grid $K_{\min} = 1$ and $K_{\max} = 59$ planes and the Cartesian $L_{\min} = 1$ plane defined the Y constant symmetry plane. New K_{\min} , K_{\max} and L_{\min} planes extending one grid spacing beyond the symmetry plane were generated using GRIDED; a NASA Ames grid manipulation software package. The amended grid sizes were $51 \times 61 \times 35$ for the slender-body grid and $50 \times 30 \times 22$ for the Cartesian grid.

D. PEGSUS IBLANK PROCEDURE

The outer boundary of the slender-body was enclosed by the Cartesian grid, except for the coincident Y-symmetry plane. Information was passed between the grids using the protocol dictated in the interpolation stencil. Redundant grid points were located in the overlapping region enclosed by the slender-body. The PEGSUS input file contained the locations of the slender-body boundaries and the points to be blanked out in the Cartesian grid. The input file is attached as Appendix A and is annotated with comments describing the individual parameters.

1. Boundary Definitions

Several options were available to help define the blanked or holed out regions within a mesh. Used most frequently were the hole boundary and outer boundary surface definitions.

a. Outer Boundary

The outer boundary of the enclosed mesh was defined using the J, K, L coordinate system to define the surfaces. One or more surfaces formed an outer boundary.

Only the outer surface through which another grid would receive information in the computational domain was included as an outer boundary.

Figure 7 shows the red $L_{\max}=35$ outer boundary for the slender-body. Information passed directly through this surface to the Cartesian grid. The outer boundary was defined using the single L_{\max} surface.

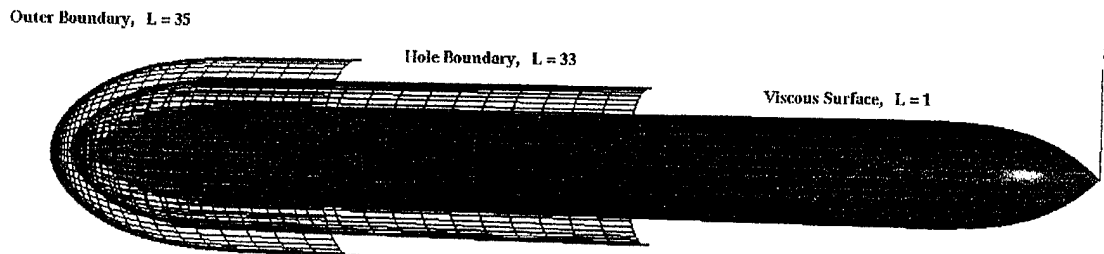


Figure 7. Slender Body Surface Representation of the Hole and Outer Boundary

b. Hole Boundary

The hole boundary defined the beginning of the hole-out region. Every hole boundary included a normal vector which was defined to point away from the holed-out region. Areas common to all specified surfaces and their normals were holed. The potential hole-out area specified with an open region extended infinitely. The closing of the region was accomplished with intersecting surfaces. Figure 8 shows areas identified for holing using open and closed formats.

The blue surface in Figure 7 (L=33) was the hole boundary. The normal pointed in the positive L direction and identified the hole boundary.

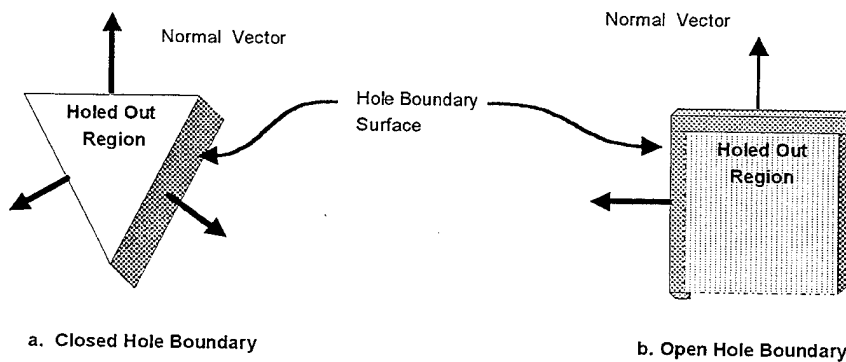


Figure 8. Representation of Closed and Open Hole Boundary Definitions

2. Orphan Point Elimination

Elimination of orphan points was an iterative process. It was important to fully comprehend what created an orphan point and to determine the best method of elimination of these points.

a. Interpolation Stencil Requirements

Each mesh in the multi-block system received information from adjacent meshes through its outer boundary or hole boundary. A user-defined priority list predetermined which meshes could communicate with each other. These linked meshes were donor meshes and provided the interpolation stencils for the boundary points. A valid interpolation stencil required none of the donor mesh grid points be a hole or boundary point. Coincident grid points formed a valid stencil if they were neither hole nor boundary points. Figure 9 shows the relation between valid and invalid stencils. Points A and B have a hole boundary point within the stencil making it invalid, point C is valid since it is coincident with a corner point of the donor mesh and point D has a valid stencil.

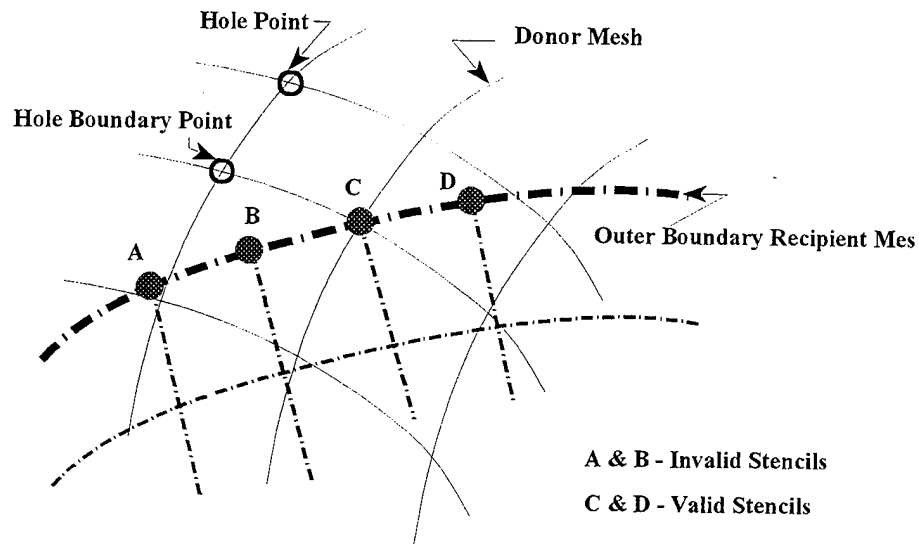


Figure 9. Valid and Invalid Interpolation Stencils [Ref. 10]

b. Elimination Techniques

PEGSUS produced an output file which identified the quality of the interpolation stencils it was able to generate. The orphan points were tabulated and identified by location and mesh. Grid viewing tools such as FAST [Ref. 12] and PLOT3D [Ref. 13] were used to plot the hole and outer boundaries of the meshes against the orphan points. Inspection revealed insufficient overlap in these regions. The planes of the donor mesh, enclosing the orphans, were recorded which allowed exact grid modifications.

PEGSUS provided controls for the translation and rotation of meshes through a global assignment in the input file (see Appendix A). Limited orphan points could be eliminated quickly by shifting the donor mesh to provide the required overlap. Care had to be taken to retain the symmetry planes and solid body orientations.

The most efficient method of ensuring adequate overlap was by selective placement of grid points during initial grid generation. The inner body grid was first constructed with a set density and outer boundaries. The Cartesian grid required at least two points between the curved outer-body boundary and hole boundary. GRIDGEN was used to record the X,Y,Z location of the body boundary extremes which included the fore and aft apexes and the mid-body. Cartesian-grid control points were placed at the recorded locations and subsequent spacing set to ensure overlap.

E. OVERFLOW ANALYSIS

The flow analysis over the slender-body was computed first with an inviscid Euler analysis and then a viscous thin-layer Navier-Stokes analysis. This approach provided immediate feedback on the symmetry of the grid and validity of the input file boundary conditions. The OVERFLOW input file for the viscous solution is attached as Appendix B.

1. Input Conditions

The Euler and Navier-Stokes solutions were initiated by treating the body surface as either an inviscid or viscous adiabatic wall. The free-stream flow was initialized at a Mach

number of 0.8 and a Reynold's number of 1.67×10^5 . These were applied to the Cartesian outer boundaries. Symmetry boundary condition was applied to the XZ plane and axisymmetry was applied to the body stings.

The ARC3D diagonal scheme was chosen as the solution method. This was the fastest scheme in per-iteration CPU time and typically the most robust. A time step of one and minimum CFL of zero were initially selected. A unit time step should be tried initially then reduced until the solution shows some form of convergence [Ref. 14].

2. Results

OVERFLOW produced a solution file, *q.save*, a residual file, *resid.out*, and a force and moment file, *fomo.out*, which were required to evaluate the CFD model. Two restarts of the solution file were run using iterations of 150 and 3000. The solution was checked for convergence by ensuring the residual (or L_2 norms) of the density had decreased by at least two orders of magnitude. Symmetry was checked by plotting the lift coefficient which was negligible at zero degrees angle-of-attack. MATLAB script files, written to quickly plot the residual and force and moment values at each iteration, are attached as Appendix C. Figure 10 is the MATLAB plot of the L_2 norm of the density for the viscous solution.

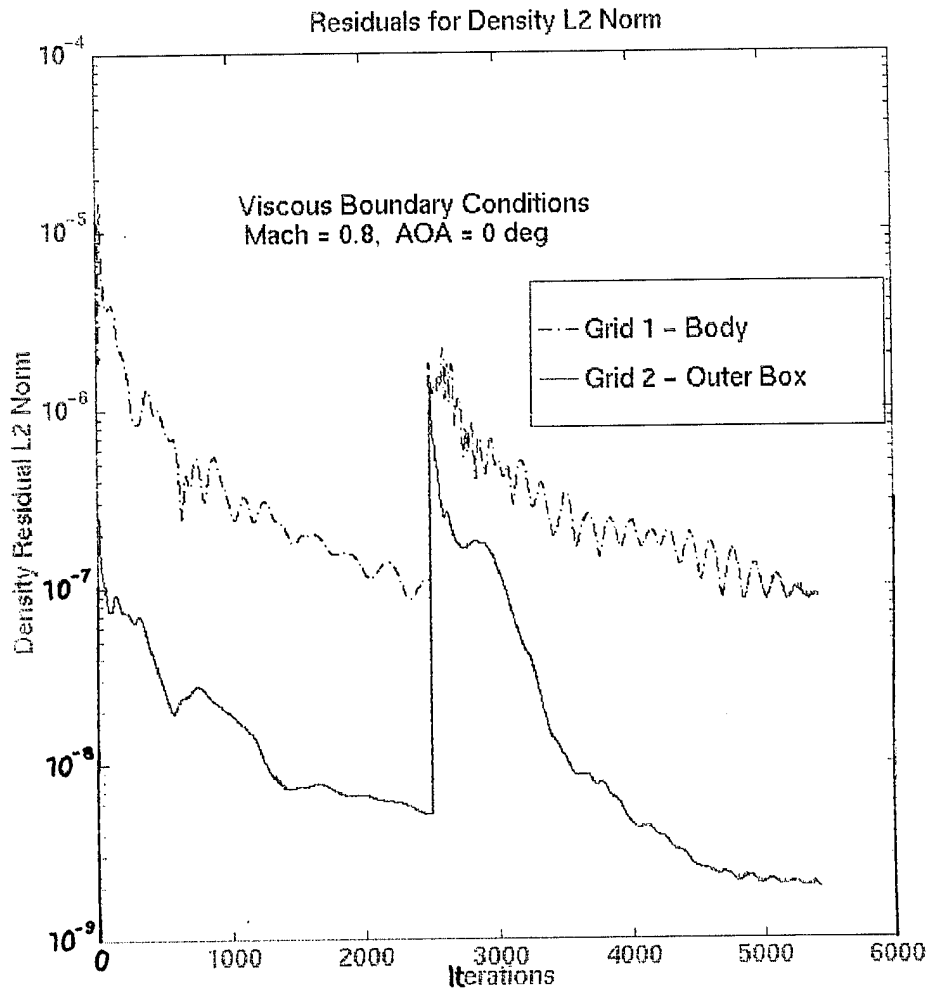


Figure 10. Density Residual L_2 Norm for the Viscous Slender-Body Flowfield Solution

Figure 11, produced using FAST [Ref.12], shows the Mach contours around the slender-body. Symmetry of the solution is evident in the contours which were plotted on the XZ plane. Maximum flow acceleration occurred, as expected, over the forebody (flow from left to right). Tangential inviscid flow at the solid boundaries was verified by plotting the velocity vectors.

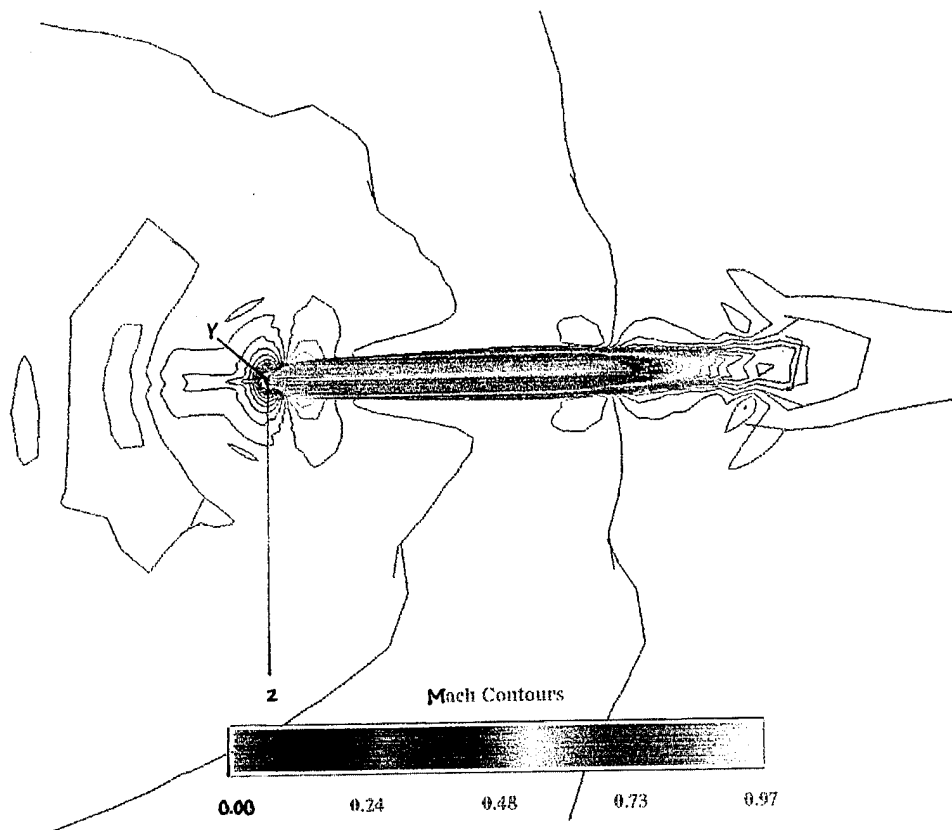


Figure 11. Mach Contour Plot of the Slender-Body Flow-field Solution

IV. BODY AND WING ANALYSIS

A. OVERVIEW

The slender-body solution procedure was expanded to include a symmetric delta-wing, as displayed in Figure 3. Five grids totalling over a million grid points were required to model the flowfield. The slender-body grid was compressed and regenerated, a C-grid was used around the wing, the wing/body junction required a collar grid and two Cartesian grids completed the structure.

Several combinations of wing/body grids were attempted. The objective was to obtain a flexible procedure which would allow the interchange of grid meshes. The resulting grid method allowed for changing wing geometries with a limited amount of complementary changes to the Cartesian and body grids.

Figure 12 shows selected surfaces of the wing/body multi-block grid scheme. The green outer Cartesian grid displays the extended plane beyond symmetry required by OVERFLOW. The body and red inner Cartesian grid were extended in a similar manner. The Chimera overlapping is apparent in the two Cartesian grids and the body/collar/wing surface grids. The holed out region of the inner Cartesian grid surrounds the body surface grid.

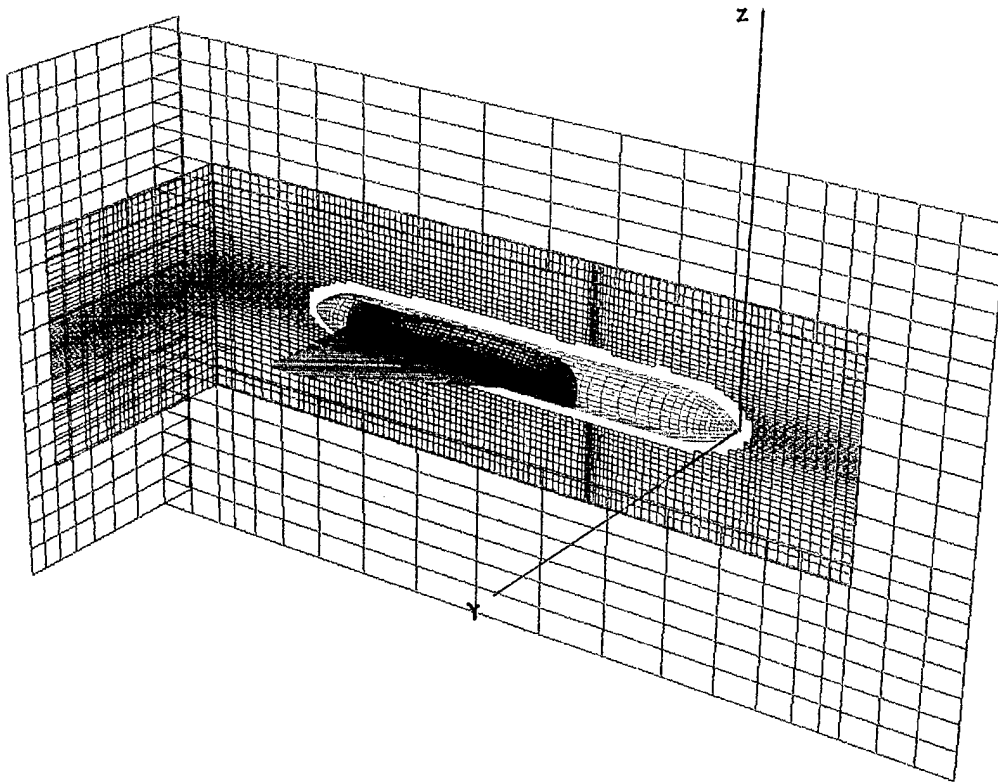


Figure 12. Selected Surfaces of the Wing/Body Overlapping Grid Structure

B. GRID GENERATION

Both GRIDGEN and HYPGEN were used to generate grids. HYPGEN provided an efficient method of building a volume grid from a surface grid when the final volume shape was insignificant. The HYPGEN input files are attached as Appendix D.

1. Body Grid

The body grid was compressed and dimensions were decreased to $91 \times 32 \times 32$. The K points were halved as the volume was marched out 0.5 diameters vice 3 diameters. The surface grid was generated in GRIDGEN and HYPGEN was used to march out the volume grid. The HYPGEN input file specified the initial and final grid spacing, J and K boundary conditions, the marching distance and the smoothing parameters. The least amount of smoothing while ensuring positive volumes and Jacobians produced the best grid.

2. Wing Grid

A $249 \times 40 \times 30$ C-grid, generated in GRIDGEN, was used to define the delta wing which had an aspect ratio of 2.31, a thickness ratio of 0.05 and leading edge sweep of 60 degrees. The exit planes of the grid were extended thirty-five points aft of the trailing edge to account for the wake region. The wing grid extended through the body and originated at the Y-symmetry plane.

Figure 13 shows the wing grid profile. The wing L grid points were stretched out geometrically with an initial spacing of 8.0×10^{-5} to a height of 0.3 diameters. The outer boundary was kept within the slender-body's solid surface to accommodate the collar grid overlap region. The delta-wing had a node representing the sharp tip. This region became very dense with converging grid planes creating large aspect ratios for the inner most volumes. It was necessary to keep the initial spacing in the L direction as well as the trailing edge spanwise spacing tight.

Wing C-Grid Planes - Red Lmax = 30, Blue K = 14 and 40

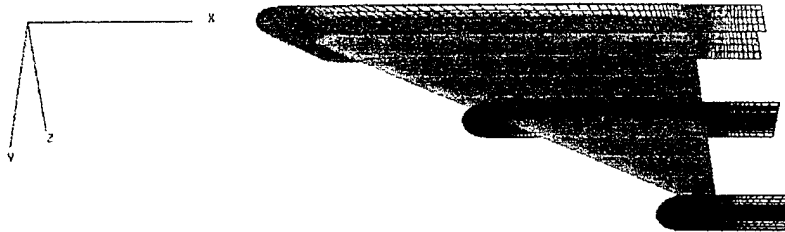


Figure 13. Wing 249 x 40 x 30 C-Grid

3. Collar Grid

The collar grid provided a smooth and effective means of communication between the intersecting body and wing grids. The collar grid was generated using portions of the COLLAR generation routines written by Chiu and Chan of NASA Ames [Ref. 15]. The following procedure was used to create the collar grid:

1. create the intersecting line between the wing and body;
2. create the surface of the collar which was coincident with the body surface;

3. create the surface of the collar which was coincident with the wing surface;
4. concatenate the two surface halves to create a total surface collar grid; and
5. march the grid in the L direction away from the wing/body junction using HYPGEN.

The 'UNIX Make File' attached in Appendix D contains the commands to generate a collar grid using the COLLAR routine. Copies of the necessary input files are also included.

a. Wing/Body Intersecting Line

The intersecting line was created using the COLLAR routine option one (create an intersecting line). The J grid lines of the wing were selected as the intersecting index which ensured 249 intersection points.

b. Body Conforming Collar Surface

Option two of the COLLAR routine generated a conforming surface grid from the intersection line and the shape of the body. An input file was required which specified the grid spacing, marching distance and smoothing parameters. Smoothing the surface grid became a trade-off between orthogonality of grid lines in the wake region and smooth rounding about the sharp leading edge of the intersection line. Figure 14 was the best obtainable output. The sharp leading edge volumes were skewed and it was necessary to move several of the grid points manually in GRIDGEN to correct the problem. Figure 15 shows the corrected leading edge portion of the body collar grid.

c. *Wing Conforming Collar Surface*

The second half of the collar surface grid was generated in GRIDGEN. The COLLAR routine option three could create a surface conforming grid on the wing; however the stretching methods used did not handle highly swept planforms satisfactorily. Figure 16 was the result of the COLLAR routine. Note the difference in stretching distances on the leading and trailing edges. The COLLAR routine input file gave minimal control over the final grid shape which resulted in the skewed collar surface grid. Growing a collar volume grid from the skewed surface grid was not possible.

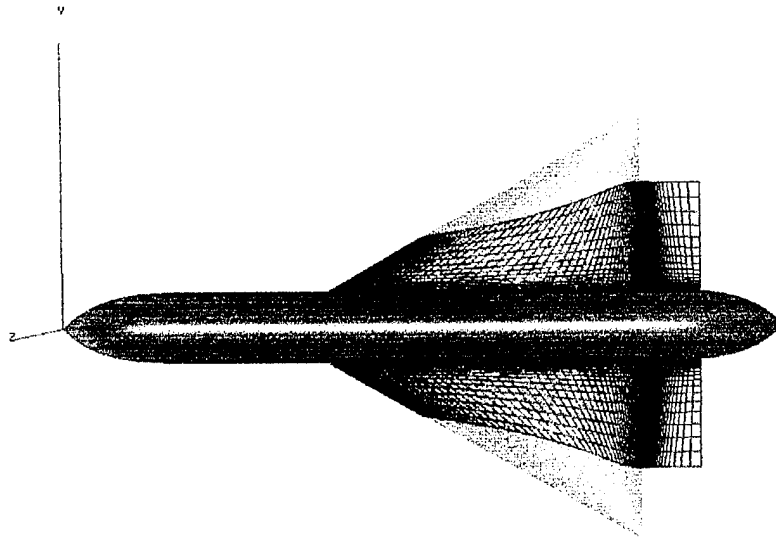


Figure 16. The Skewed COLLAR Routine Output of the Wing Conforming Collar Surface

The GRIDGEN approach produced an excellent wing collar half. The intersection line was read into GRIDGEN as a connector from which a full wing surface, $249 \times 30 \times 1$, was generated. GRIDED was then used to extract a subset of the wing, $249 \times 22 \times 1$. The extracted wing was now the desired shape of the second collar half. Figure 17 shows the results of the GRIDGEN method of producing the wing conforming collar surface grid. The wing collar grid lines were coincident with the wing grid which simplified the PEGSUS holing out procedure.

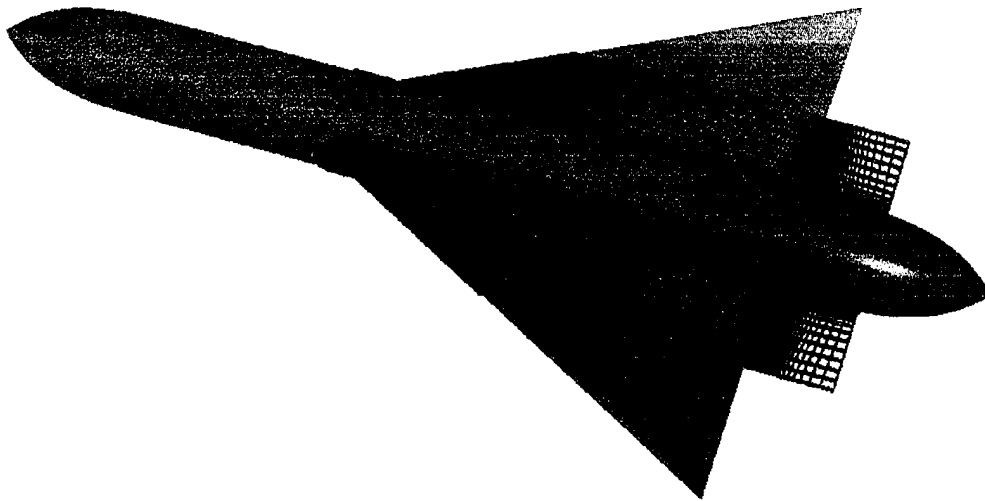


Figure 17. GRIDGEN Produced Wing Conforming Collar Surface Grid

d. Surface Grid Concatenation

The body and wing collar halves were concatenated using GRIDED. It was necessary to ensure the J and K grid lines ran in similar directions to yield a right-handed continuous collar surface grid.

e. Volume Grid Generation

HYPGEN was used to march the collar grid 0.6 diameters away from the body. The final collar grid dimension was $249 \times 41 \times 32$. The collar grid outer boundary had to extend beyond both outer boundaries of the wing and body. The sharp concave intersection required an initial grid spacing of 1.0×10^{-5} to ensure a smooth growth. A smoothing exponent of five was necessary to handle growth around the sharp leading edge. The smoothing exponent was most strongly affected by the number of L grid planes and the initial grid spacing. It was essential that, during construction of the collar surface halves, the first grid spacing away from the intersection line was equal on either side. Failure to do so led to unequal growth at the intersection line resulting in negative volumes and Jacobians. Figure 18 shows the collar and wing grid outer boundaries which ensured the growth of the collar beyond the outer boundary of the wing.

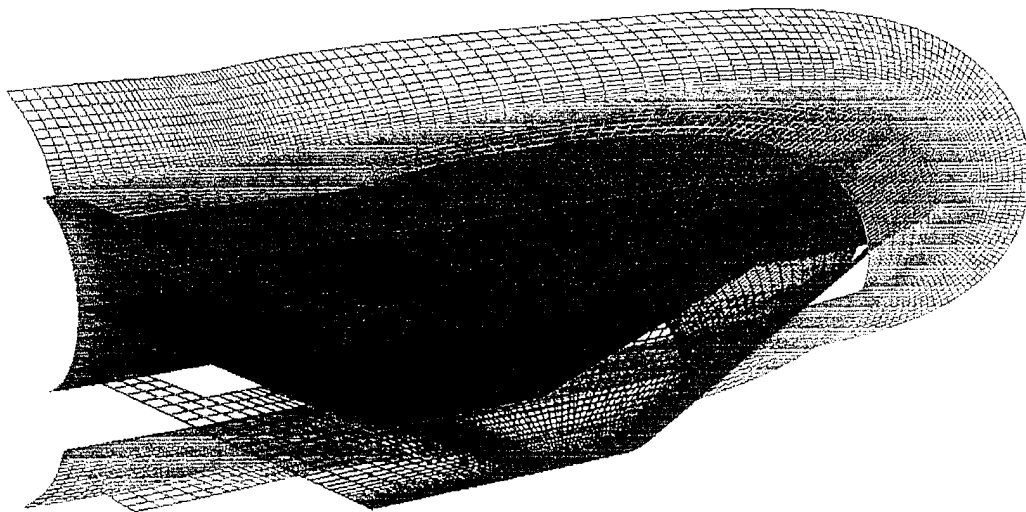


Figure 18. The Outer Boundaries of the Wing and Collar Grids

4. Cartesian Grids

The wing, body and collar grids were enclosed by an inner Cartesian grid, $100 \times 50 \times 66$, which was enclosed by a second outer Cartesian grid, $60 \times 50 \times 31$. Both grids had a common Y-symmetry plane which was extended one L grid plane using GRIDED . The inner Cartesian grid was sufficiently dense to provide the necessary overlap of the wing, body and collar.

C. PEGSUS IBLANK PROCEDURE

The iblanking procedure did not change from the two grid model; however, it was more complex to ensure sufficient overlap of the five grids. The wing and body grids created holes in each other and the inner Cartesian grid. A void was left that was filled by the collar grid. The collar grid formed the solid boundary of the wing-body intersection.

Iblanking difficulties were greatest around the outer boundaries of the convex body, the swept convex leading edge and the convex/concave collar. Repositioning of points on the Cartesian grid eliminated most orphan points. Also moving the hole boundaries inward one plane increased the overlap which further reduced the orphan points; however, there were then more redundant points to be computed and an increase in CPU time would be expected. The PEGSUS input file is attached in Appendix A.

D. OVERFLOW ANALYSIS

The composite grid with iblank information and interpolation stencils along with the input file were used by OVERFLOW to compute both inviscid and viscous solutions. The inviscid solution provided confidence in the selected boundary conditions and symmetry of the grid. A selected OVERFLOW input file for the viscous model is attached in Appendix B. Only the viscous solutions will be presented.

1. Boundary Conditions

The body, wing and collar solid surfaces were treated as viscous adiabatic walls. Both the wing and collar wake regions were assigned the C-grid flow-through boundary conditions. Only one side of the wake region was specified in the input file but this was applied equally to both sides. The free-stream characteristic condition was applied to the outer Cartesian grid. This imposed free-stream conditions, then applied the supersonic/subsonic inflow/outflow condition, holding pressure fixed for subsonic outflow and applying a characteristic condition for subsonic inflow [Ref. 14]. Symmetry boundary conditions were applied to the two Cartesian grids and the body grid on the XZ plane.

2. Turbulence Modeling

The Baldwin-Lomax algebraic models were used to compute the turbulent eddy viscosity within the flowfield. Both models searched for a maximum of a turbulence function $F(y)$, dependant on local vorticity, which was denoted as F_{\max} [Refs. 14 and 16]. Regions were specified to contain the search for F_{\max} . A limiting value called the Degani-Schiff cutoff was implemented which stopped the search for F_{\max} when $F(y)$ dropped below the product of the cutoff and the current F_{\max} . The cutoff was specified via the TLPAR1 parameter in the input file.

a. *Baldwin-Lomax Boundary Layer Model*

The standard model, option 1, was applied to the wall bounded regions. This model was the easiest to use as no extra equations were solved for which convergence had

to be monitored. The selected turbulent wall regions were the body, wing and collar $L = 1$ surfaces. The J and K start and end parameters (JTLS/E and KTLS/E) specified the bounds of the wall region. A Degani-Schiff cutoff of 30% , $TLPAR1=0.3$, was applied as suggested when separation in the shear layer was not important [Ref. 14].

b. Baldwin-Lomax Shear Layer Model

The shear layer model, option 11, was applied to wake regions. The ITDIR parameter specified the L direction perpendicular to the shear layer and the JTLS/E etc. identified the region for which the shear layer model set the turbulent eddy viscosity. The shear-layer model was applied to the wake region of the wing and collar grids. No Degani-Schiff cutoff was applied, $TLPAR1 = -1$, as suggested for the standard Baldwin-Lomax model [Ref. 14]. The shear layer formulation is further explained in Appendix G of Reference 14.

3. Results

Validation of the quality of the wing/body grid scheme and flow-solver was done by comparing the results with experimental wind tunnel data for a similar geometry. Oelkner, Bergmann and Hummel completed such measurements (Ref. 11) with thirty pressure tap locations along the body and wing. Results including lift coefficients and static pressure coefficients were published for various angles-of-attack (AOA) and sideslip at a free-stream Mach number of 0.3. Comparisons were performed between computed and experimental

results for zero sideslip and AOAs of 0° , 8.7° and 19.3° . The streamwise viscous terms in the J direction were turned off for 0° and 8.7° where separation was unlikely and turned on for 19.3° AOA solution. The OVERFLOW parameters that were varied to produce a converged solution are listed in Table 1.

OVERFLOW Parameters	0 deg	8.7 deg				19.3 deg		
Iterations	500	100	2000	2000	2000	100	2000	2000
Dt (time step)	0.1	0.1	0.4	0.6	1	0.1	0.1	0.4
CFL min	0.005	0.5	0.5	0.5	0.5	0.5	0.5	0.5
DIS 4 (4th ord dissipation)	0.04	0.04	0.04	0.08	0.12	0.04	0.04	0.04

Table 1. Various OVERFLOW Parameters for the Computed Solutions

Plots of the density L_2 norm, the pressure lift coefficient and pitching moments are attached as Appendix E. OVERFLOW used the individual wall region area (the viscous surface identified in the input file) as the reference area and a unit length for the force and moment coefficient computations. The lift coefficients for the three runs were plotted against the wind tunnel data in Figure 19. The OVERFLOW lift coefficient computation was corrected to the wing reference area for comparisons. The 0° and 19.3° results were very

close to the published data; however, at 8.7° there was a difference of 24% in the coefficient values. The lift curve slopes of the wind tunnel data and a fitted trend line were within 5%. Comparisons of additional flow characteristics and pressure distributions follow.

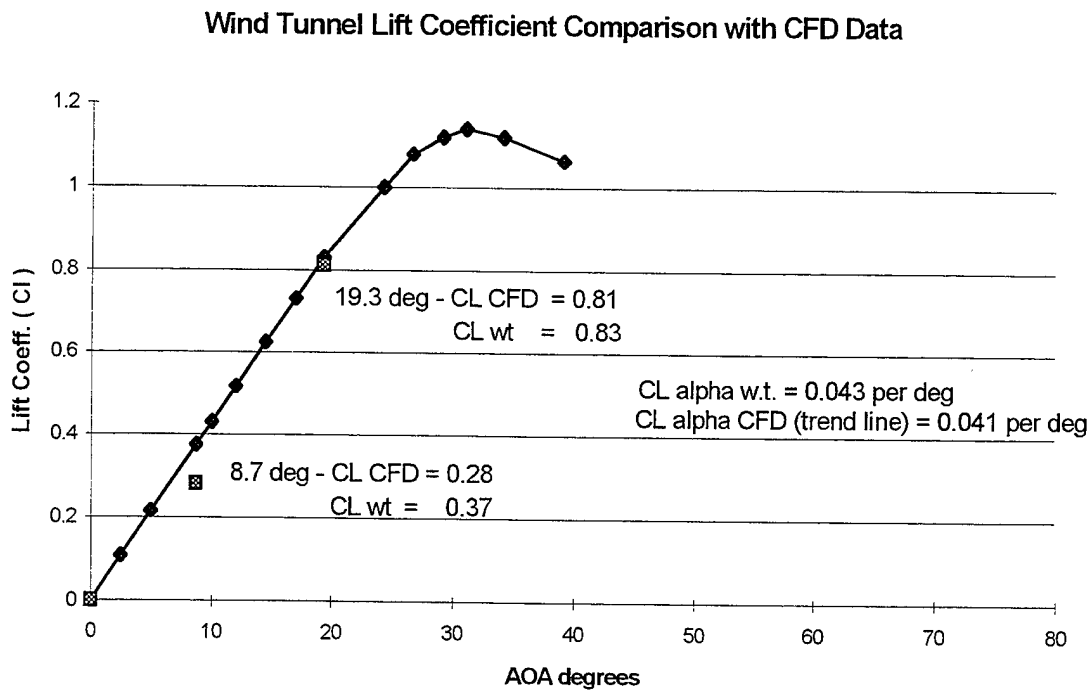


Figure 19. A Comparison of Published Wind Tunnel Data (Ref. 11) with the CFD Solution.

a. Zero Degrees AOA

Solution symmetry was first checked by plotting the Mach contours at zero degrees AOA for a viscous solution. Together with a negligible lift coefficient the grids were

considered to be of sufficient quality to continue the comparisons. Figure 20 shows the symmetric Mach contours and maximum velocities located over the upper and lower surfaces of the fore body.

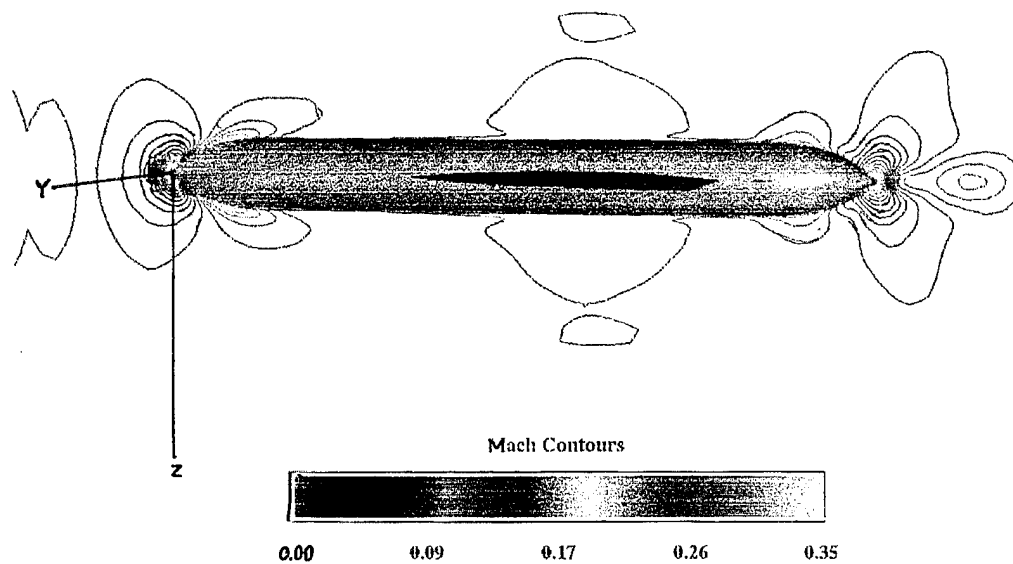


Figure 20. Mach Contours Over the Wing/Body at 0° AOA

b. 8.7 Degrees AOA

The L_2 density norm decreased over two orders of magnitude and the pressure lift coefficient had stabilized after 6000 iterations which was an indication of a converged solution. Figure 21 shows the decreasing L_2 density norm on a linear-log scale. The spikes in the data were solution restart points indicating a change in parameters. Figure 22 is the pressure lift coefficient which is initially quite oscillatory but stabilizes to a discrete value for each wall region.

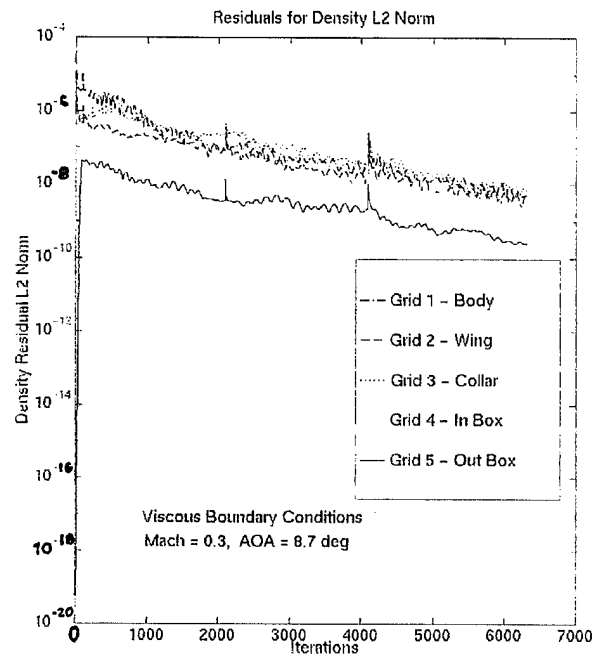


Figure 21. Convergence of the L_2 Density Norm for the Wing/Body at 8.7° AOA

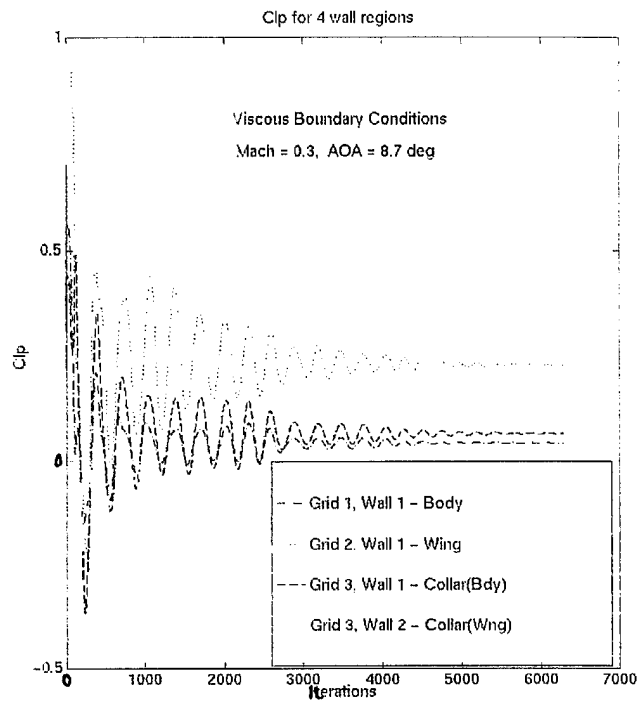


Figure 22. Pressure Lift Coefficient for the Wing/Body at 8.7° AOA

The wing/body Mach number contours are displayed in Figure 23. The maximum Mach number of approximately 0.4 occurred over the wing leading edge near the root and over the upper surface of the fore body.

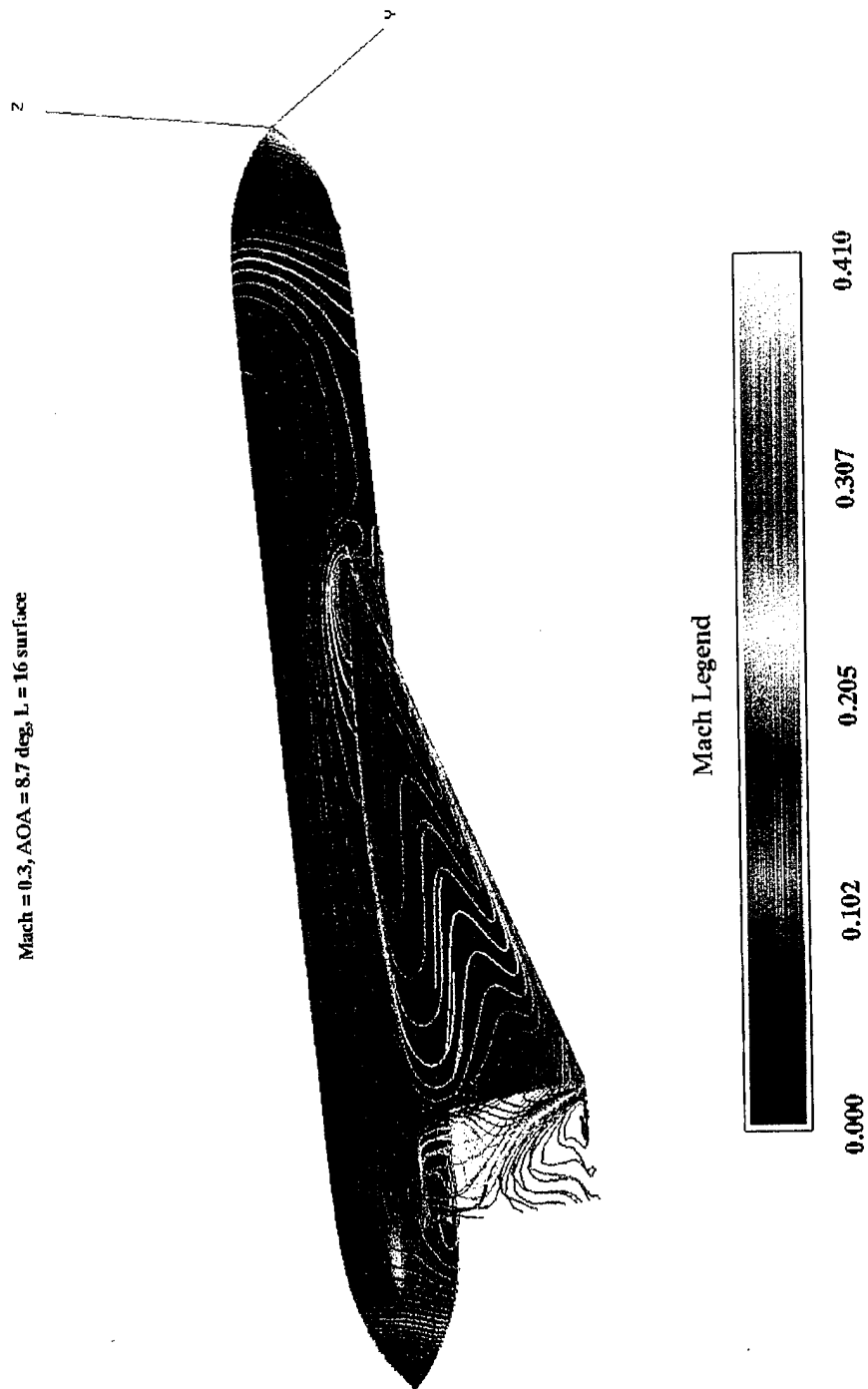


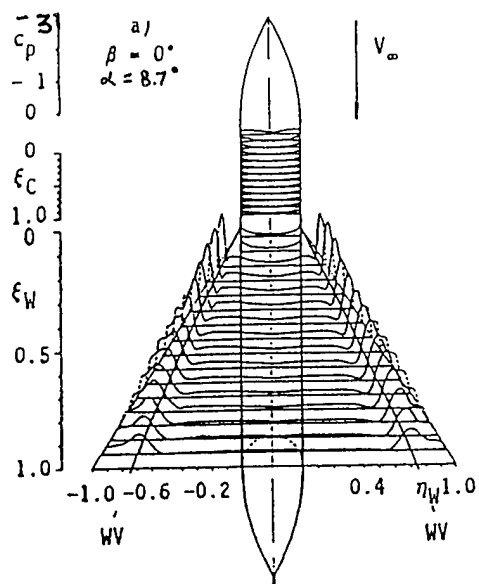
Figure 23. Wing/Body Mach Contours at 8.7 ° AOA

The static pressure coefficient (C_p) figure from reference 11 was reproduced and is shown along side that of the CFD pressure coefficient output. The horizontal lines on the reference plot Figure 24a were the locations of the static pressure ports. The value of the C_p suction peaks were determined by measuring the distance from the horizontal lines to the applicable C_p curve and comparing it with the C_p scale ranging from 0 to -3. The max suction peak value was approximately -1.5. The single line on the wing surface labeled 'WV' was the location of the wing leading edge wing vortex.

Figure 24b showed the CFD C_p contours. The maximum suction peak followed the leading edge and identified the location of the wing vortex. Maximum suction was at the leading edge wing root, identified by dark green, and this had a C_p magnitude of approximately -1.5. Note also the smooth transition of contours between the body, collar and wing grids.

Visualization of the leading edge vortex was feasible with the FAST particle trace utility and is presented in Figure 25. The flow was seeded forward of the wing root at the fourth and eighth planes away from the body (magenta and blue particle traces respectively). The vortex location matched the C_p suction peak profile.

A.



B.

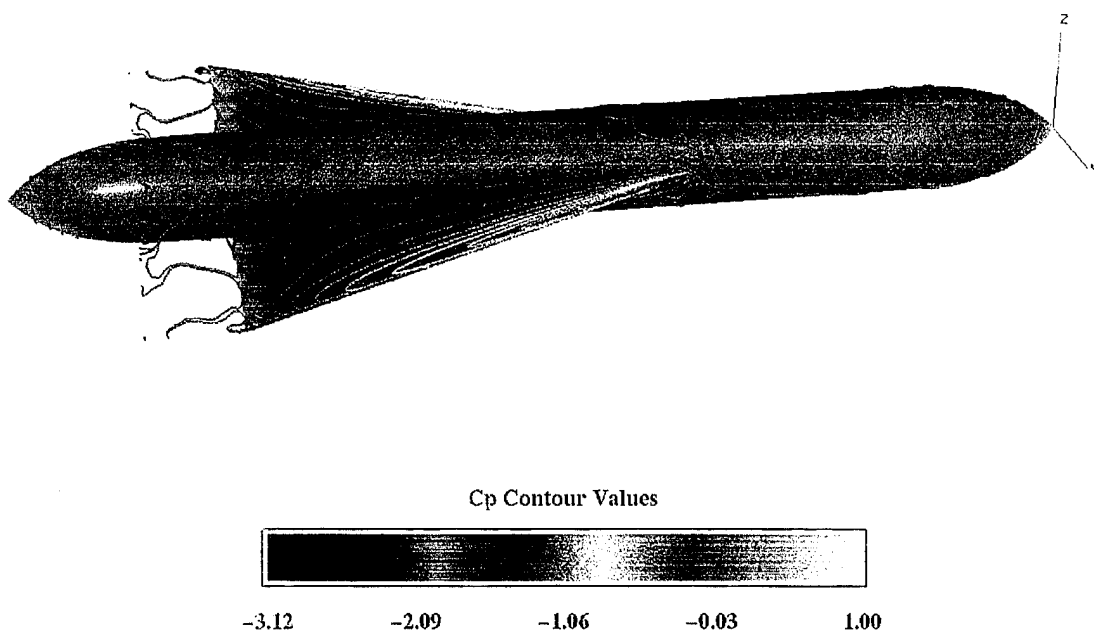


Figure 24 a. Wind Tunnel C_p Plot (Ref 11.) , b. CFD C_p Contour Plot at 8.7° AOA

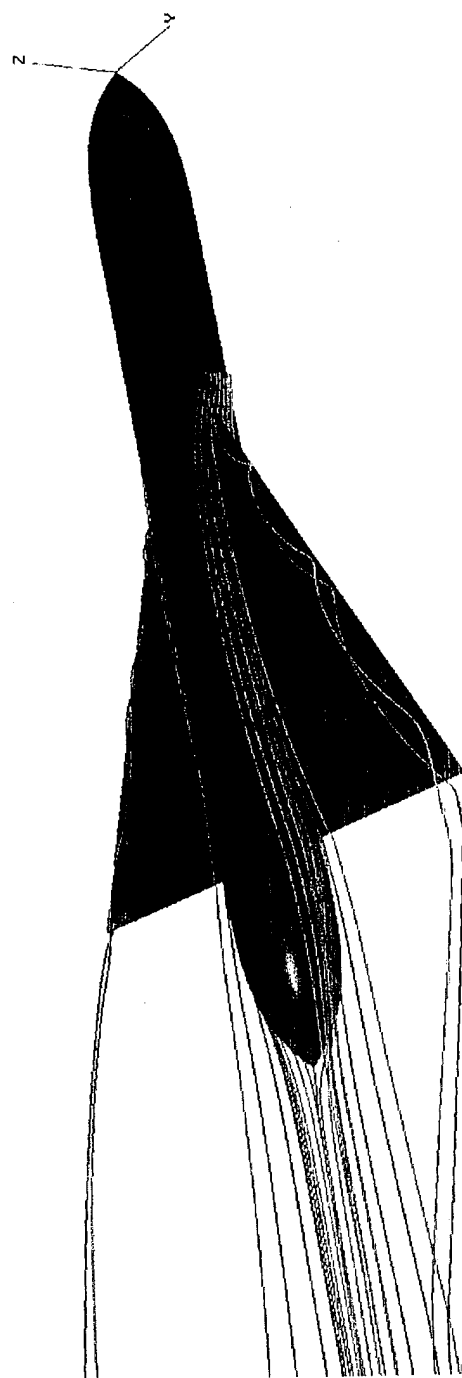


Figure 25. Wing/Body Particle Trace at 8.7° AOA

c. 19.3 Degrees AOA

The Mach number contours at the increased AOA, displayed in Figure 26, showed regions of sonic flow at the wing leading edge root which coincided with the maximum suction peak and vortex origin. The aft half of the leading edge indicated free-stream velocities (blue contours) and showed an inboard shift of the Mach number peaks when compared with the Mach number contours for the 8.7° AOA test case.

The C_p contours at 19.3° AOA were similarly compared against the published data in Figure 27. The maximum suction peak values for both plots were approximately -4.0 and were located at the leading edge root where the flow approached sonic conditions. Both figures indicated a 20% inboard shift of the leading edge vortex. Once again note the smoothness of the contour lines over the three grid boundaries.

Figure 28 shows the velocity vectors highlighted with C_p values at a plane close to the wing tip region. The vector density was been reduced for viewing. A tight vortex could be seen about the leading edge where the velocity vectors were reversing direction. The boundary layer shape was evident close to the trailing edge as was the typical wake region profile.

Figure 29 is a particle trace identifying the location of the vortex and its early separation from the wing. The blue path lines show the flow sweeping off the body and onto the wing.

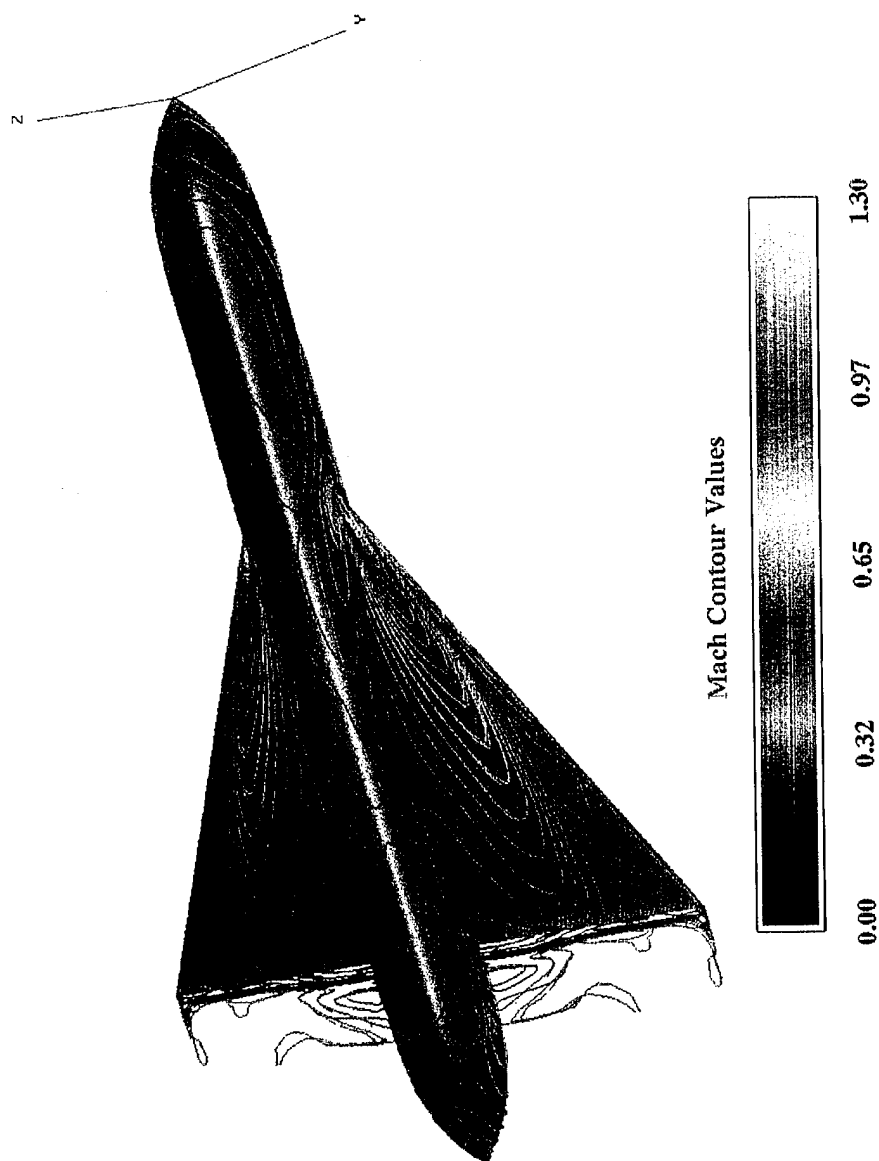
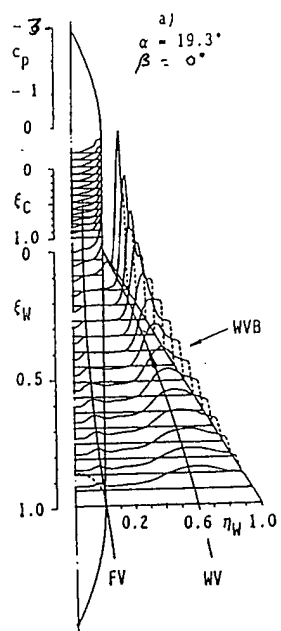


Figure 26. Wing/Body Mach Contours at 19.3° AOA

A.



B.

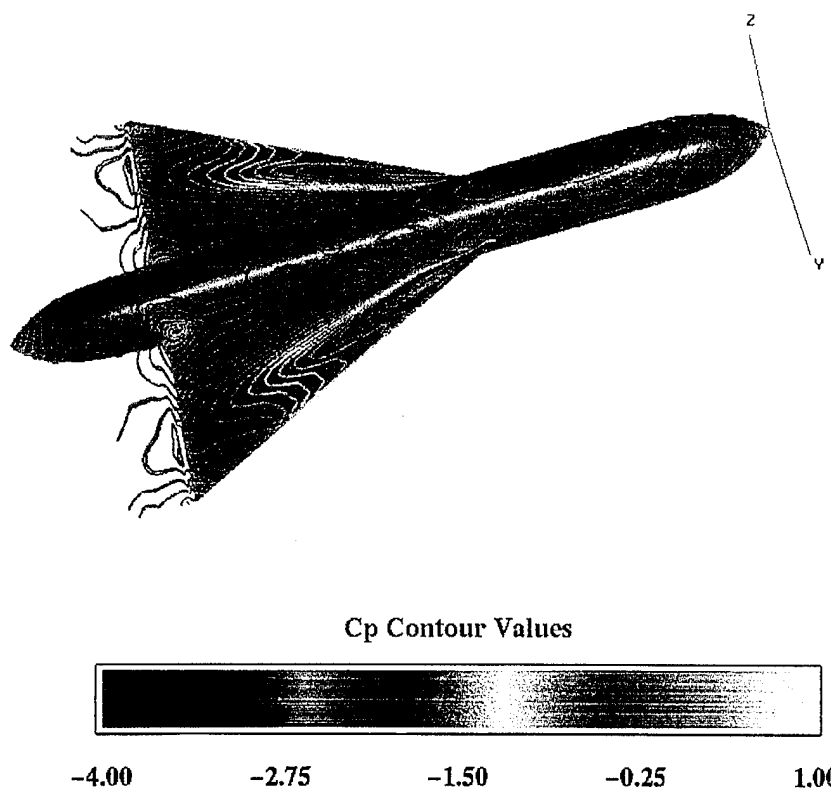


Figure 27 a. Wind Tunnel C_p Plot (Ref. 11) , b. Wing/Body C_p Contours at 19.3° AOA

(Note - Vector Density Reduced for Improved Viewing)

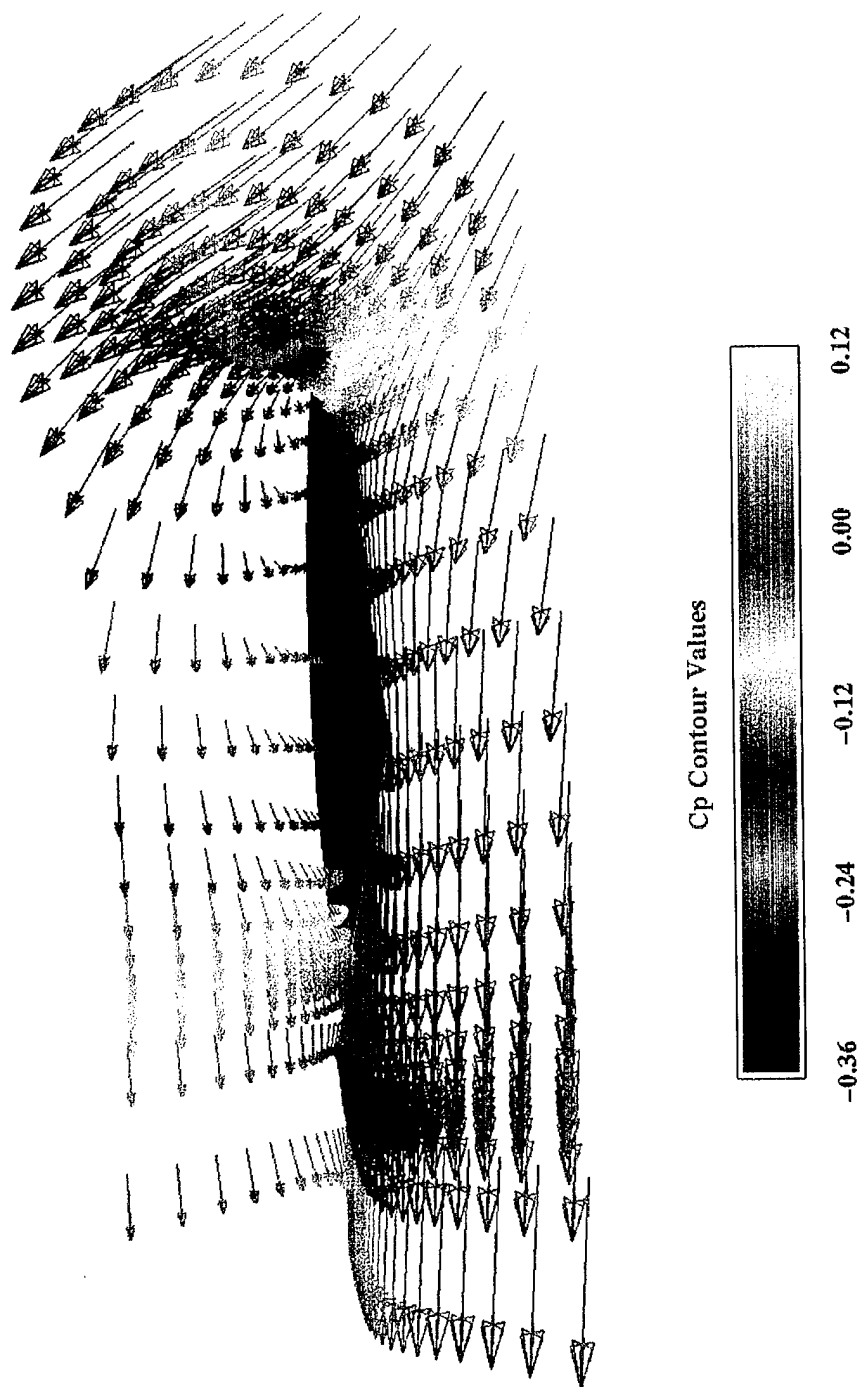


Figure 28. Wing Tip Velocity Vector Profile at 19.3° AOA

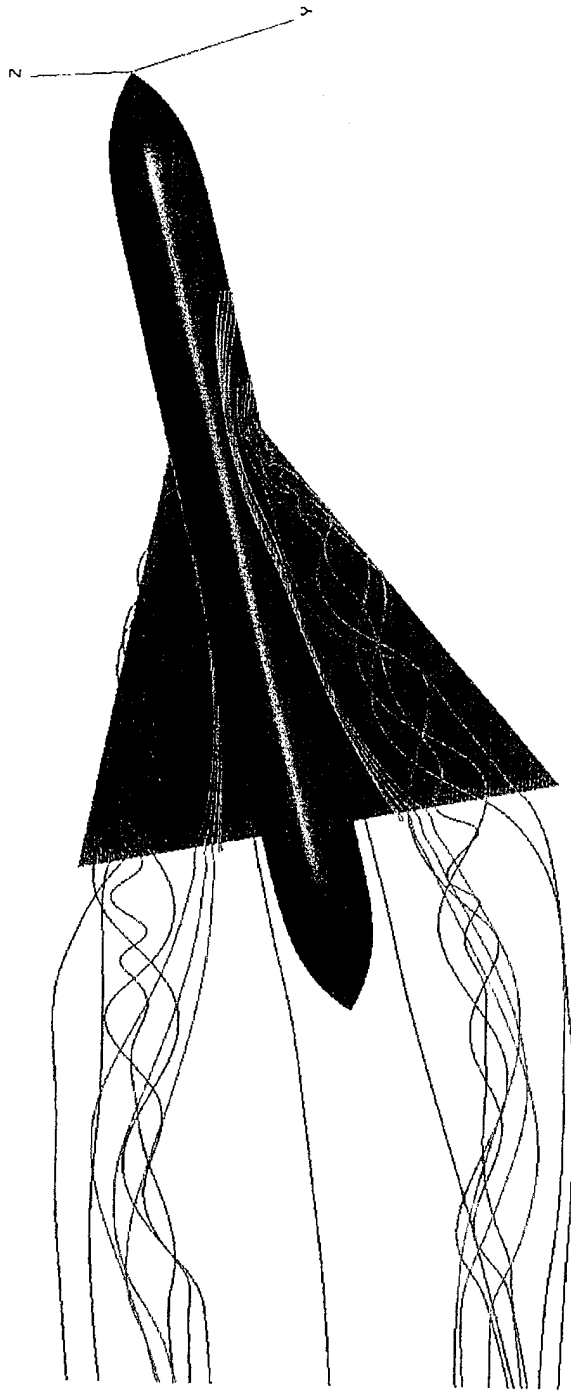


Figure 29. Wing/Body Particle Traces at 19.3° AOA

V. CONCLUSIONS AND RECOMMENDATIONS

The multi-block CFD procedure was established and successfully applied to a missile geometry. The requirements of the CFD software were established and the installation of the software on the CRAY was validated. Both the elliptic and hyperbolic grid generation methods were used and evaluated. A successful method was developed for creating a collar grid around a highly swept thin-wing. A procedure for improving grid overlap, by specifying coincident planes at the hole and outer boundaries when possible, was established. A wing C-grid was successfully built around sharp leading and trailing edges and a single point tip. The CFD results were compared to published wind tunnel measurements (Ref. 11) and are in good agreement with the lift and pressure data, vortex formations and subsequent development.

Learning the multi-block procedure was best achieved by starting with a simple two grid overlapping scheme, completing the CFD multi-block process flowchart and then applying the understood procedure to a more complicated geometry.

Questions often arose during the analysis as to the quality, symmetry, etc., of a given grid. It was useful to isolate the single grid and compute a single grid solution for comparison with the multi-block solution. This eliminated the use of RMG2PEG, PEGSUS and MERGE thus minimizing possible sources of error.

Hyperbolic grids were developed by marching grid planes away from the supplied surface grids. Smoothing parameters and initial/final grid spacing were available; however, there was minimal control over the internal grid point spacing and the outer boundary shape. The elliptic grids required the outer boundary shapes be specified. Controls were available to specify grid point clustering, orthogonality and smoothing. Geometries with sharp convex or concave surface shapes, such as supersonic airfoils, were best modelled using the elliptic grid generator GRIDGEN. The hyperbolic grid generator (HYPGEN) worked most effectively on smooth symmetric grids such as missile bodies and collars.

The wing/body geometry required five overlapping grids and a million grid points. Grid dimensions were dictated by the complexity of the surface geometry. The slender-body had concentrated grid points along the fore and aft sections ensuring a unit aspect ratio at the apex and smooth stretching ratios. Initial boundary layer grid spacing was dependent on the flow-solver turbulent model.

The intersection of the wing and body was modelled by a collar grid. The collar provided the best method of ensuring adequate wing/body overlap. Generating a collar surface grid around a highly swept wing required using the COLLAR routine, to generate the intersection line and the body collar surface and GRIDGEN to create the wing collar surface.

The inner Cartesian grid density ensured a smooth transition from the highly dense wing/body/collar grids to the less dense outer Cartesian free-stream grid. Cartesian grid point clustering around the wing tip was necessary as the wing C-grid did not extend beyond the tip.

The overlapping multi-block approach (Chimera) required the smooth interpolation of data at the grid boundaries. The accuracy of the solution was directly dependant on the quality of the interpolation stencils. PEGSUS identified regions requiring grid refinement through the use of orphan points. The best procedure for ensuring adequate overlap of grid boundaries differed for the two grid generators. The HYPGEN input file allowed specification of several grid parameters. Overlap of the body and collar grids was best accomplished by assigning similar initial/final grid spacing, marching distances and smoothing factors. GRIDGEN provided specific grid point placement capabilities. During the grid generation, prior knowledge of the PEGSUS boundaries was used to ensure grid planes of the outer overlapping grid were coincident with the expected boundary planes of the inner grid. Though coincident boundary/plane overlap was not required it did yield the highest quality stencils and fewer orphan points.

The wing/body CFD solutions were computed and compared to the experimental data (Ref. 11). The thin-layer Navier-Stokes solution was used at low angles of attack (0° and 8.7°) when minimum separation was present. At higher incidence angles the viscous terms in all the directions were included. The CFD lift coefficient data agrees with

the experimental results at 0° and 19.7° and was within 25% at 8.7° . The lift curve slopes were within 5% when a trend line was passed through the CFD data. More CFD information at various AOAs is required to better define the lift curve characteristics. The locations and magnitudes of the wing suction peaks compared favorably. As the AOA was increased the C_p chordwise maximum suction values moved inboard from the leading edge. The location of the leading edge vortex followed the maximum C_p movement.

The wing/body grid scheme, software and multi-block procedure has been validated. An excellent project continuation would involve incorporating a canard and the computation of the close-coupling effects of reference 11. The canard would be a scaled version of the delta-wing and would entail similar grid requirements. A collar grid would be necessary. The reduced span of the canard would reduce the grid skew effects experienced during generation of the wing/body collar. It is likely that the entire collar could be generated with the COLLAR routine as opposed to the established GRIDGEN-COLLAR approach.

LIST OF REFERENCES

1. Kafyeke, F. and Piperni, P., "Application of the MBTEC Euler Code to the Challenger and to the CF-18 Complete Aircraft Configurations", Canadian Aeronautics and Space Journal, Vol. 39, No. 4, December, 1993.
2. Steger, J.L., Dougherty, F.C. and Beneck, J.A., "A Chimera Grid Scheme in Advances in Grid Generation", K.N. Ghia and U. Ghia, Eds., ASME FED-5, 59-69, 1983.
3. Beneck, J.A., Buning, P.G. and Steger, J.L., "A 3-D Grid Embedding Technique", AIAA Paper No. 85-1523, July 1985.
4. Buning, P.G., Chiu, I.T., Obayashi, S., Rizk, Y.M., and Steger, J.L., "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent", AIAA-88-4359-CP, AIAA Atmospheric Flight Mechanics Conference, Minneapolis, Minnesota, August, 1988.
5. Fan, Y.S., "An Investigation of the Transonic Viscous Drag Coefficient for Axisymmetric Bodies", Master of Science in Aeronautical Engineering, Naval Postgraduate School, March, 1995.
6. Ekaterinaris, J.A., "Computation of Flowfields Over Missile Configurations", AIAA-94-1915, AIAA 12th Applied Aerodynamics Conference Paper, Colorado Springs, Colorado, June, 1994.
7. Reuter W.H., "Flowfield Computations Over the Space Shuttle Orbiter with a Proposed Canard at a Mach of 5.8 and 50 Degrees Angle of Attack", Master of Science in Aeronautical Engineering, Naval Postgraduate School, March, 1992.
8. Steinbrenner, J.P., and Chawner, J.R., "GRIDGEN Release Notes Version 9.6", MDA Engineering, Arlington, Texas, October, 1994.
9. Chan, W.M., Chiu, I.T. and Buning, P.G., "User Guide for HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface Version 1.3", NASA TM 108791, NASA Ames Research Center, Moffet Field, California, January, 1993.
10. Suhs, N.E., and Tramel, R.W., "PEGSUS 4.0 User's Manual", AEDC-TR-91-8, Arnold Engineering Development Center, Arnold Air Force Base, Tennessee, November, 1991.

11. Oelker, H-C, Bergmann, A., and Hummel, D. "Vortex Formation Over a Close-Coupled Canard-Wing-Body Configuration", AGARD-CP-494, 1991.
12. "Flow Analysis Software Toolkit Version 1.1a - FAST", Numerical Aerodynamics Simulation Division, NASA Ames Research Center.
13. Walatka, P.P. and Buning, P.G., "PLOT3D User's Manual Version 3.5", NASA Ames Research Center, California, 1989.
14. Buning, P.G., and others, "OVERFLOW User's Manual Version 1.6aw", NASA Ames Research Center, California, July, 1995.
15. Parks, S., Buning, P., Chan, W. and Steger, J., "Collar Grids for Intersecting Geometric Components Within the Chimera Overlapped Grid Scheme", AIAA-91-1587, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, June, 1991.
16. Baldwin, B.S. and Lomax, H., "Thin Layer Approximation and Algebraic Model for Separated Turbulent Flows", AIAA Paper 78-257, January, 1978.
17. Wendt, J.F., Anderson, J.D., Degrez, G., Dick, E., and Grundmann R., "Computational Fluid Dynamics an Introduction", Springer-Verlag, 1992.
18. Thompson, J.F., Warsi, Z.U.A., and Mastin, W.C., "Numerical Grid Generation - Foundations and Applications", Elsevier Science Publishing Co., New York, New York, 1985.

APPENDIX A. PEGSUS INPUT FILES

Nov 26 1995 15:27

```

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF COLLAR',
  J RANGE = 1,1,
  K RANGE = 1,41,
  L RANGE = 1,32,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF COLLAR',
  J RANGE = 249,249,
  K RANGE = 1,41,
  L RANGE = 1,32,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF COLLAR',
  J RANGE = 1,249,
  K RANGE = 1,1,
  L RANGE = 1,32,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF COLLAR',
  J RANGE = 1,249,
  K RANGE = 41,41,
  L RANGE = 1,32,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF COLLAR',
  J RANGE = 1,249,
  K RANGE = 1,41,
  L RANGE = 32,32,
  $END

C-----
C C INBOX GRID (100x50x66)
C-----
$BOUNDARY NAME = 'OUTER BOUNDARY OF INBOX',
  ISPARTOF = 'INBOX',
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF INBOX',
  J RANGE = 1,1,
  K RANGE = 1,50,
  L RANGE = 1,66,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF INBOX',
  J RANGE = 1,100,
  K RANGE = 1,50,
  L RANGE = 1,66,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF INBOX',
  J RANGE = 1,100,
  K RANGE = 1,1,
  L RANGE = 1,66,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF INBOX',
  J RANGE = 1,100,
  K RANGE = 50,50,
  L RANGE = 1,66,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF INBOX',
  J RANGE = 1,100,
  K RANGE = 1,50,
  L RANGE = 66,66,
  $END

```

Nov 26 1995 15:27

```

K RANGE = 1,48,
L RANGE = 64,64,
H VOUT = '+L',
$END

C-----
C BOUNDARY DEFINITIONS
C-----
C These are the outer boundary definitions through which other meshes
C receive information.
C-----

C C BODY GRID (91x32x32)
C-----
$BOUNDARY NAME = 'OUTER BOUNDARY OF BODY',
  ISPARTOF = 'BODY',
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF BODY',
  J RANGE = 1,91,
  K RANGE = 1,32,
  L RANGE = 32,32,
  $END

C-----
C C WING GRID (249x40x30)
C-----
$BOUNDARY NAME = 'OUTER BOUNDARY OF WING',
  ISPARTOF = 'WING',
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF WING',
  J RANGE = 1,1,
  K RANGE = 1,40,
  L RANGE = 1,30,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF WING',
  J RANGE = 249,249,
  K RANGE = 1,40,
  L RANGE = 1,30,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF WING',
  J RANGE = 1,249,
  K RANGE = 1,1,
  L RANGE = 1,30,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF WING',
  J RANGE = 1,249,
  K RANGE = 40,40,
  L RANGE = 1,30,
  $END

$SURFACE ISPARTOF = 'OUTER BOUNDARY OF WING',
  J RANGE = 1,249,
  K RANGE = 1,40,
  L RANGE = 30,30,
  $END

C-----
C C COLLAR GRID (249x41x32)
C-----
$BOUNDARY NAME = 'OUTER BOUNDARY OF COLLAR',
  ISPARTOF = 'COLLAR',
  $END

```

```

      RANGE = 33, 33,
      NROUT = 2, 1, $END

      SUBROUTINE
      -----
      These are the outer boundary definitions through which other meshes
      receive information.

      $BOUNDARY NAME = 'OUTER BOUNDARY OF BODY',
      ISPARTOF = 'BODY', $END

      $SURFACE ISPARTOF = 'OUTER BOUNDARY OF BODY',
      RANGE = 2, 50,
      KRANGE = 2, 60,
      LRANGE = 35, 35, $END

```

```
C PEGSUS INPUT FILE - SLENDER BODY
C
C By : Bret Barton
C Date : November 26, 1995
C .....
C .....
C General Information
C .....
C This input file is used with the grid file 'INGRID' by PEGSUS 4.0
C to create the interpolation stencil used by OVERFLOW. The output files
C COMPOUT (composite mesh) and IAPLOT (IAPL files concatenated
C in program MERGE44 into 'grid.in'. This file and PEGSUS interpolation
C information file 'IPTOUT' are required by OVERFLOW.
C .....
C Note - Start typing in second column!!!
C .....
SGLOBAL
QUALITY = 0.9,0.29,-0.3,
EPS      = 0.005,
          SEND
C .....
C Exclude boundary points (via the INCLUDE statement) on the downstream
C boundary, reflected symmetry planes, or surfaces since these points will
C be initialised directly by OVERFLOW boundary conditions.
C .....
C The X0, Y0 and Z0 are translation coordinates which sometimes are
C useful to eliminate small numbers of Orphan Points. Rotation is also
C available.
C .....
C Grid Dimen - BODY (51x61x35), OUTBOX (50x10x21)
C .....
SHESH NAME = 'BODY',
LINK       = 'OUTBOX',
JINCLUDE   = 2.50,
KINCLUDE   = 2.60,
LINCLUDE   = 2.35,
X0 = 0.0,
Y0 = 0.0,
Z0 = 0.0,
          SEND
C .....
SHESH NAME = 'OUTBOX',
LINK       = 'BODY',
JINCLUDE   = 2.49,
KINCLUDE   = 2.29,
LINCLUDE   = 2.20,
X0 = 0.45,
Y0 = 0.0,
Z0 = .1,
          SEND
C .....
HOLE DEFINITIONS
C .....
These are the hole boundaries which identify grid points to be
eliminated from the computed solution. The location of the boundary
ensures sufficient overlap for a valid interpolation stencil.
C .....
$BOUNDARY NAME = 'HOLE IN OUTBOX MADE BY INBODY',
            ISPARTOF = 'BODY',
            HOLEIN = 'OUTBOX',
            SEND
C .....
$SURFACE ISPARTOF = 'HOLE IN OUTBOX MADE BY BODY',
                KRANGE = 1.51,
                KCRANGE = 1.61,
```


Nov 26 1995 15:27

peg41.inp

Page 3

```

SEND
SSURFACE ISPARTOF = 'HOLE IN BODY/INBOX MADE BY WING',
  JRANGE = 4,125,
  KRANGE = 4,4,
  LRANGE = 1,27,
  NVOUT = '-K',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/INBOX MADE BY WING',
  JRANGE = 125,246,
  KRANGE = 4,4,
  LRANGE = 1,27,
  NVOUT = '-K',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/INBOX MADE BY WING',
  JRANGE = 4,125,
  KRANGE = 4,37,
  LRANGE = 27,27,
  NVOUT = '+L',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/INBOX MADE BY WING',
  JRANGE = 125,246,
  KRANGE = 4,37,
  LRANGE = 27,27,
  NVOUT = '+L',
  SEND

C-----
C COLLAR GRID (249x41x32)
C-----

BOUNDARY NAME = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  ISPARTOF = 'COLLAR',
  WHOLEIN = 'BODY','WING','INBOX',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 246,246,
  KRANGE = 5,38,
  LRANGE = 1,29,
  NVOUT = '+J',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 4,4,
  KRANGE = 5,38,
  LRANGE = 1,29,
  NVOUT = '-J',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 4,125,
  KRANGE = 38,38,
  LRANGE = 1,29,
  NVOUT = '+K',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 125,246,
  KRANGE = 38,38,
  LRANGE = 1,29,
  NVOUT = '+K',
  SEND

```

Nov 26 1995 15:27

peg41.inp

Page 4

```

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 4,125,
  KRANGE = 5,5,
  LRANGE = 1,29,
  NVOUT = '-K',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 125,246,
  KRANGE = 5,5,
  LRANGE = 1,29,
  NVOUT = '-K',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 4,125,
  KRANGE = 5,38,
  LRANGE = 29,29,
  NVOUT = '+L',
  SEND

SSURFACE ISPARTOF = 'HOLE IN BODY/WING/INBOX MADE BY COLLAR',
  JRANGE = 125,246,
  KRANGE = 5,38,
  LRANGE = 29,29,
  NVOUT = '+L',
  SEND

C-----
C INBOX GRID (100x50x66)
C-----

BOUNDARY NAME = 'HOLE IN OUTBOX MADE BY INBOX',
  ISPARTOF = 'INBOX',
  WHOLEIN = 'OUTBOX',
  SEND

SSURFACE ISPARTOF = 'HOLE IN OUTBOX MADE BY INBOX',
  JRANGE = 3,3,
  KRANGE = 3,48,
  LRANGE = 1,64,
  NVOUT = '-J',
  SEND

SSURFACE ISPARTOF = 'HOLE IN OUTBOX MADE BY INBOX',
  JRANGE = 98,98,
  KRANGE = 3,48,
  LRANGE = 1,64,
  NVOUT = '+J',
  SEND

SSURFACE ISPARTOF = 'HOLE IN OUTBOX MADE BY INBOX',
  JRANGE = 3,98,
  KRANGE = 3,3,
  LRANGE = 1,64,
  NVOUT = '-K',
  SEND

SSURFACE ISPARTOF = 'HOLE IN OUTBOX MADE BY INBOX',
  JRANGE = 3,98,
  KRANGE = 48,48,
  LRANGE = 1,64,
  NVOUT = '+K',
  SEND

SSURFACE ISPARTOF = 'HOLE IN OUTBOX MADE BY INBOX',
  JRANGE = 3,98,

```


APPENDIX B. OVERFLOW INPUT FILES

Dec 8 1995 15:23

```

$NITERS SEND
$METRM SEND
$TIMACU SEND
$SMOACU SEND
$VSWINP
      VISCL = F., VISCL = .T., VISCL = F.,
      STORB = 0,
      $END
$BCINP
      NBC = 6,
      IBTYP = 12, 12, 32, 32, 32, 32,
      IBDIR = 3, 1, 1, 2, -2, -3,
      JBCE = 1, 1, 1, 1, 1, 1,
      JBCE = -1, 1, 1, -1, -1, -1,
      KBCE = 1, 1, 1, 1, 1, 1,
      KBCE = -1, -1, -1, 1, 1, -1,
      LBCE = 1, 1, 1, 1, 1, 1,
      LBCE = -1, -1, -1, -1, -1, -1,
      $END
$SCINP SEND

```

```
C OVERFLOW INPUT FILE - SLENDER BODY & CARTESIAN GRID
C
C By : Bret Barton
C Date: November 26, 1995
C
C *****
C General Information
C *****
C
C This is the first input file for a viscous solution. 100 steps are initially
C run after which alpha and DT parameters are increased, RESTART=.T. and
C NSTEPS increased to 1000 (this system specific). The 'resid.out'
C file produces an L2 norm of the solution density which helps establish
C convergence.
C
C Viscosity in the J indices is turned off due to the low angle of attack
C (thin layer H-Stokes solution) and Jaon at high AOA.
C The Baldwin-Slimak Shear Layer turbulence model was used on the solid
C body surfaces.
C *****
C $GLOBAL
C NSAVE=100, NSTEPS=100, RESTART=.F., NSAVE = 100,
C NGROWTH = 0,
C SEND
C *****
C SFLOWIMP
C FSNACH=C-8, ALPHA = 0.00, REY = 0.166667E6, TINF = 576.,
C SEND
C $VARGAM SEND
C
C $GRIDNAM
C GRIDNAME = 'BODY',
C SEND
C SHITERS
C SHITERS
C SEND
C SMETPRN
C IHRS = 0, ILHS = 2, IDISS = 3,
C SEND
C STIMACSH
C ITIME = 1, DT = 0.1, CFLMIN= 0.5,
C SEND
C $$$OCU
C ISPEC = 2, DIS2 = 10.00, DISA = 0.4,
C SNOC = 0.00,
C SEND
C $VISINP
C VISCK = .F., VISCK = .T., VISCL = .T.,
C IHTYP = 1,
C ITDIR = 1,
C JTLS = 1,
C JTLE = -1,
C KTLS = 1,
C KTLB = -1,
C LTLS = 1,
C LTLE = -1,
C TLPAR1=0.3,
C SEND
C $BCINP
C NBC = 5,
C IBTP = 5, 15, 12, 12, 15,
C IBDR = +3, +1, +2, -2, -1,
C JBCE = 1, 1, 1, 1, -1, -1,
C KBCE = -1, -1, -1, -1, -1, -1,
C LBCE = -1, -1, -1, -1, -1, -1,
C LCBE = 1, 1, 1, 1, 1, 1,
C LBCE = 1, -1, -1, -1, -1, -1,
C SEND
C $$$ENCHP
C ENCHP
C SEND
C
C $GRIDNAM NAME = 'OUTBOX',
C SEND
```


NTURBB = 0,		NTURBB = 1,	
SEND	\$BC\$INP	SEND	\$BC\$INP
NBC	1	NBC	1
IBTYP	12	IBTYP	12
IDBIN	3	IDBIN	3
JBCS	-1	JBCS	-1
KBCS	-1	KBCS	-1
LBCS	-1	LBCS	-1
LACE	1	LACE	1
\$SC\$INP	\$END	\$SC\$INP	\$END

NTURBB = 0,		NTURBB = 1,	
SEND	\$BC\$INP	SEND	\$BC\$INP
NTURBB	0,	NTURBB	0,
IBTYP	12	IBTYP	12
IDBIN	3	IDBIN	3
JBCS	-1	JBCS	-1
KBCS	-1	KBCS	-1
LBCS	-1	LBCS	-1
LACE	1	LACE	1
\$SC\$INP	\$END	\$SC\$INP	\$END

APPENDIX C. MATLAB SCRIPT FILES

```

end
end
end
%
% fomo_3.out
%
if Nfiles >= 3
    load fomo_3.out;
    Data3 = fomo_3;
    Clear fomo_3
    Count=0;
    Mrows = size (Data3 , 1);
    % initialise counter
    % number of rows
    % separates the grid information into specific vectors
    for I3 = I3+1 : I3*Mrows/Nwall,
        for J = 1 : Nwall,
            Count=Count+1;
            Coef(I3,J) = Data3(Count,Colm);
        end
    end
end
end
%
% fomo_4.out
%
if Nfiles == 4
    load fomo_4.out;
    Data4 = fomo_4;
    Clear fomo_4
    Count=0;
    Mrows = size (Data4 , 1);
    % initialise counter
    % number of rows
    % separates the grid information into specific vectors
    for I4 = I3+1 : I3*Mrows/Nwall,
        for J = 1 : Nwall,
            Count=Count+1;
            Coef(I4,J) = Data4(Count,Colm);
        end
    end
end
end
%
% OUTPUT
% Plot the requested coefficient for each wall region.
% This plot statement will change depending on the number of wall
% regions (viscous/inviscid surfaces) in the individual solution.
%
% (m,n)=size(Coef);
% length=length(m);
% plot (Length,Coef(1,1),'-', 'Length,Coef(:,2)', '-','Length,Coef(:,3)', '--','Length,Coef(:,4));
%
% assign titles and labels to plot
%
if XX ==1
    title(['Cdp for ', num2str(Nwall) ' wall regions ',' ], ylabel('Cdp'), xlabel('Iterat
converged')
elseif XX ==2
    title(['Cdp for ', num2str(Nwall) ' wall regions ',' ], ylabel('Cdp'), xlabel('Iterat
ons');
else
    title(['Cdp for ', num2str(Nwall) ' wall regions ',' ], ylabel('Cdp'), xlabel('Iterat
ons');
endif
endif
endif

```

```

MATLAB FORCE & MOMENT PLOTTING ROUTINE

Program "readfomo.m"
Bret Barton 23 May 95

This MATLAB file reads the OVERFLOW force and moment file " fomo.out "
The fomo.out data will be automatically loaded into MATLAB.

File fomo.out is a history of the force and moment coeff for each
viscous or inviscid wall of each grid. Columns list the GRID ,
WALL #, STEP #, Clp, Cdp, Cyp, Clf, Cdf, Cyf, Cl, CM, CN, AREA.

clear; % clear all variables

User enters the number files & column of the data that is
desired.

files = input ('How many input files to plot (max = 4) ');
wall = input ('How many viscous or inviscid walls ? ');
answer = 1;
while Answer == 1, % allows several coeff to be plotted
    disp ('
    The following numbers represent coefficients which can be plotted ');
    K = input ('Enter the number of the coefficient to be plotted. ');
    Colmn = KK';
    while Answer == 1, % allows several coeff to be plotted
        disp ('
        The following numbers represent coefficients which can be plotted ');
        K = input ('Enter the number of the coefficient to be plotted. ');
        Colmn = KK';
    end
end

fomo_1.out % fomo_1.out

% filename of data to be plotted

% number of rows in fomo.out
% counter

separates the grid information into specific vectors

for I1 = 1 : Rows/Nwall,
    for J = 1 : Nwall,
        Counts=Count+1;
        Coef(I1,J) = Data1(Count,Colmn);
    end
end

% filename of data to be plotted

% number of rows in fomo.out
% counter

separates the grid information into specific vectors

for I2 = 1 : I1+Rows/Nwall,
    for J = 1 : Nwall,
        Counts=Count+1;
        Coef(I2,J) = Data2(Count,Colmn);
    end
end

```

```

Nov 26 1995 15:18 readfomo.m Page 3
title('Cyp for ' num2str(Nwall) ' wall regions ')) , ylabel('Cyp'), xlabel('Iterat
ions');
elseif XX ==4
title('Clf for ' num2str(Nwall) ' wall regions ')) , ylabel('Clf'), xlabel('Iterat
ions');
elseif XX ==5
title('Cdf for ' num2str(Nwall) ' wall regions ')) , ylabel('Cdf'), xlabel('Iterat
ions');
elseif XX ==6
title('Cyf for ' num2str(Nwall) ' wall regions ')) , ylabel('Cyf'), xlabel('Iterat
ions');
elseif XX ==7
title('Cl for ' num2str(Nwall) ' wall regions ')) , ylabel('CL'), xlabel('Iteratio
ns');
elseif XX ==8
title('CN for ' num2str(Nwall) ' wall regions ')) , ylabel('CN'), xlabel('Iteratio
ns');
else XX ==9
title('CN for ' num2str(Nwall) ' wall regions ')) , ylabel('CN'), xlabel('Iteratio
ns');
end
% The legend has been hard coded with values for the grid and wall
% taken from the 'fomo.out' file.
legend('Grid 1, Wall 1 - Body','Grid 2, Wall 1 - Wing','Grid 3, Wall 1 - Collar(Bdy
)', 'Grid 3, Wall 2 - Collar(Wng)');
gtext('Viscous Boundary Conditions');
%
Answer = input('Do you want to plot another coefficient? 1= Yes, 2= No ');
if Answer == 1
figure;
end
end
% END WHILE loop

```

```

Count=Count+1;
Res(I2,J) = Data2(Count,Colm);

end
end

%-----
%      resid_3.out
%-----

if Nfiles >= 3
    load resid_3.out;
    Data3 = resid_3;
    clear resid_3;
    Count=0;
    Mrows = size (Data3,1);
    % Initialise counter
    % number of rows

    % separates the grid information into specific vectors
    for I3 = I2+1 : I2 + Mrows/Mgrids,
        for J = 1 : Ngrids,
            Count=Count+1;
            Res(I3,J) = Data3(Count,Colm);
        end
    end
end

%-----
%      resid_4.out
%-----

if Nfiles >= 4
    load resid_4.out;
    Data4 = resid_4;
    clear resid_4;
    Count=0;
    Mrows = size (Data4 , 1);
    % Initialise counter
    % number of rows

    % separates the grid information into specific vectors
    for I4 = I3+1 : I3 + Mrows/Mgrids,
        for J = 1:Ngrids,
            Count=Count+1;
            Res(I4,J) = Data4(Count,Colm);
        end
    end
end

%-----
%      OUTPUT
%-----

% The plot and legend statements have been hard coded with values
% for 5 grids and will have to be changed according to the number
% of individual meshes.

% plot the residual Lq2 norm for the given data
[m,n]=size(Res);
Length=(1:m)';
semilogy (Length,Res(:,1),'-','Length,Res(:,1)');
title('Residuals for Lq2 Norm '); xlabel('Lq2 norm'), ylabel('Iterations');

```

```

%-----
%      MATLAB RESIDUAL PLOTTING ROUTINE
%-----

% Program "readresid.m"
% Bret Barton 23 May 95

%-----
%      This MATLAB file reads the OVERFLOW "resid.out"
%      file and plots the delq L2 norm for each grid.
%-----

% The resid.out data will be automatically loaded into MATLAB.
% Ensure that the "resid.out" files produced by OVERFLOW
% have been renamed to resid_1.out, resid_2.out, etc. A Maximum
% of four residual files can be plotted.

%-----
%      User enters the number of resid.out files, the number
%      of grid meshes and the column to be plotted. Column 8
%      is the L2 Density Norm but any column may be plotted.
%-----

clear;
disp('-----');
disp('This program plots the residuals from the resid.out file');
disp('produced from OVERFLOW. Enter the name of the resid.out');
disp('files to resid_1.out, resid_2.out, etc. ');
disp(' ');

Mfiles = input('How many input files to plot (max = 4) ');
Ngrids = input('Enter the number of grids ');
Colm = input('Enter column of the data to be plotted (Lq2 norm = 8) ');

%-----
%      resid_1.out
%-----

if Nfiles >= 1
    load resid_1.out;
    Data1 = resid_1;
    clear resid_1;
    Count = 0;
    Mrows = size (Data1 , 1);
    % number of rows
    % counter

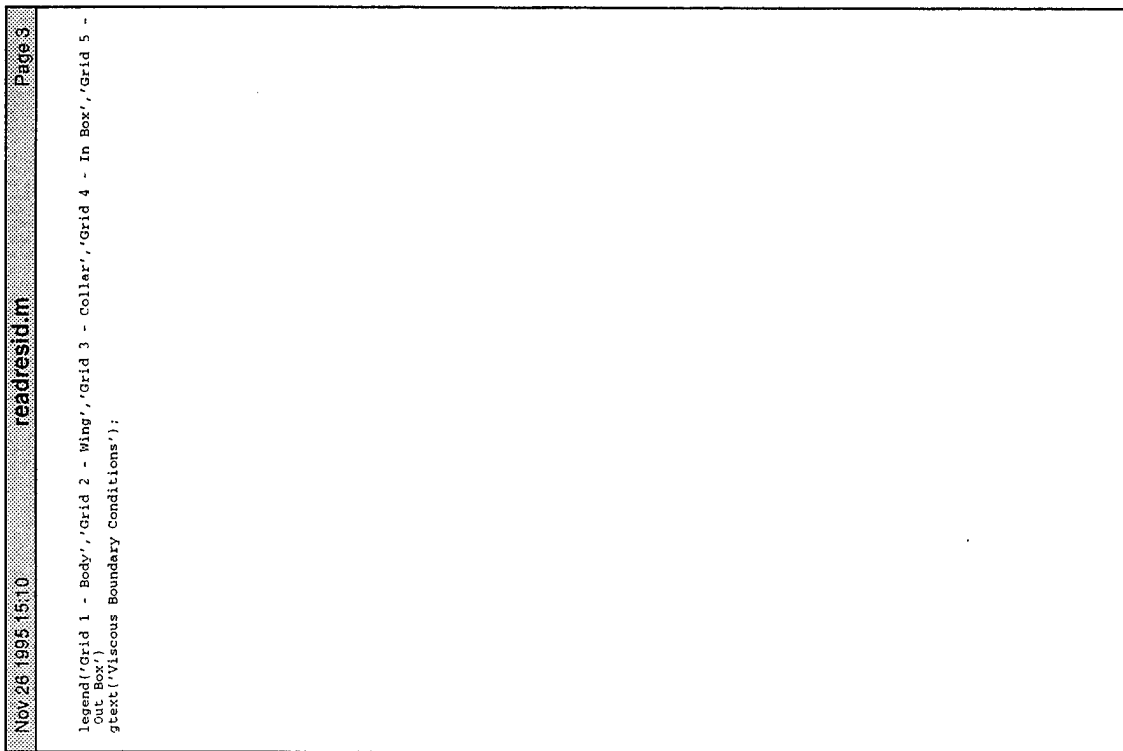
    % separates the grid information into specific vectors
    for I1 = 1 : Mrows/Mgrids,
        for J = 1 : Ngrids,
            Count=Count+1;
            Res(I1,J) = Data1(Count,Colm);
        end
    end
end

%-----
%      resid_2.out
%-----

if Nfiles >= 2
    load resid_2.out;
    Data2 = resid_2;
    clear resid_2;
    Count=0;
    Mrows = size (Data2 , 1);
    % Initialise counter
    % number of rows

    % separates the grid information into specific vectors
    for I2 = I1+1 : I1+Mrows/Mgrids,
        for J = 1 : Ngrids,

```



APPENDIX D. COLLAR AND HYPGEN INPUT FILES


```

Nov/27/1995 21:41 make_collar
# Use intgrd.com file written by collar script.
cat > plot3d_tmp.com << EOF
intgrd
quit
EOF
plot3d < plot3d_tmp.com
/bin/rm -f plot3d_tmp.com
#
# Step 5 - Grid concatenation
#
# Note - These steps may be different depending on the initial
# indexing of the surface grids. Check in PLOT3D or FAST.
#
# Concatenate the two surface grids using the GRIDED program
# directly, rather than out of the collar script. This is because
# running a program from a script with piped input like this doesn't
# work (i.e., collar would be confused after grided finished). Note
# that we could run the collar script and do the following
# interactively with no problem.
#
# We do the following steps with GRIDED:
#
# (1) read in coll.dat (part of collar on body);
# (2) reverse K direction;
# (3) concatenate col2.dat in the K-direction, assuming one
# coincident point;
# (4) reverse J direction (a result of the original line.dat);
# (5) write out the resulting grid as collar2d.dat.
#
grided << EOF
coll.dat
1 1
5 5
0 1 0
1
2012.dat
1
2 1
0
collar_2d.dat
EOF
#
# Step 6 - Generate collar volume grid in HYPGEN. The first line
# of link statement followed by the execute. Surf.i is not
# required. Collar_hypg.i is the HYPGEN input file.
#
in collar_2d.dat surf.i
hypgen < collar_hypg.i > collar_hypg.out
mv plot3d.dat collar_3d.dat
#
# Clean up.
#
/bin/rm -f surf.i

```

```

#! /bin/csh -f
#
# COLLAR GRID MAKE FILE
#
# By : Bret Barton
# Date : 5 Oct 95
#
# The following make file contains the step by step approach
# to making a collar completely with the COLLAR routine.
#
# ***Note - If the surface grids are created in GRIDGEN
# then the header must be removed.
#
# Assumptions:
#
# (1) HYPGEN has been compiled
# (2) Collar grid tools have been compiled.
# (3) PLOT3D is available.
#
# Collar surface grid.
#
# Step 1 - Run collar script to generate an intersection line.
# wing_root_2d_2dns.uf.grd = unformatted wing surface grid
# body_2d.uf.grd = unformatted body surface grid
#
./collar << EOF
1
wing_root_2d_2dns.uf.grd
body_2d.uf.grd
int_line.dat
#
# Step 2 - Run collar script to generate the body collar
# surface grid. Surbasel input file is required
#
2
surbasel
body_2d.uf.grd
coll.dat
6
EOF
#
# Check it with PLOT3D.
#
# Use surgrd.com file written by collar script.
cat > plot3d_tmp.com << EOF
surgrd
quit
EOF
plot3d < plot3d_tmp.com
/bin/rm -f plot3d_tmp.com
#
# Step 3 - Run collar script to generate the wing collar
# surface grid. Int_col input file is required.
#
3
./collar << EOF
4
int_col
int_line.dat
wing_root_2d_2dns.uf.grd
col2.dat
6
EOF
#
# Check it with PLOT3D.

```

```

Nov 27 1995 21:45      surbase1.i      Page 1
C
C      BODY COLLAR SURFACE INPUT FILE
C
C      File Name : surbase1.i
C      By       : Bret Barton
C      Date      : 5 Oct 95
C
C
C      1, 1      IBCJA, IBCJB
C      20, .6, 0.001, 0.0      KMAX, ETANX, DETA, DFEAR
C      .8, .4, 10      SHU, TIM, ITSVOL
C      0, 0, 1, 1      JSPER, KSPER(0/1), JSAXSA, JSAXSB(0/1)
C
C-----
C      INPUT NOTES:
C-----
C      Boundary condition types for J=1
C      IBCJA = -1 Free floating condition
C      = 1 Constant plane condition in X, float Y and Z
C      = 2 Constant plane condition in Y, float X and Z
C      = 3 Constant plane condition in Z, float X and Y
C      = 4 Reflected symmetry condition in X
C      = 5 Reflected symmetry condition in Y
C      = 6 Reflected symmetry condition in Z
C      = 10 Periodic condition in X
C      = 11 Floating condition along KS=const line in +JS direction
C      = 12 Floating condition along KS=const line in -JS direction
C      = 13 Floating condition along JS=const line in +KS direction
C      = 14 Floating condition along JS=const line in -KS direction
C      = 15 Floating condition along user supplied line of points
C      = 16 Exact coordinates of boundary points prescribed by user
C      = 21 Constant X plane for all J from 1 to JMAX (*)
C      = 22 Constant Y plane for all J from 1 to JMAX (*)
C      = 23 Constant Z plane for all J from 1 to JMAX (*)
C
C      (*) Must also apply for IBCJB
C
C      Similarly for IBCJB at J=JMAX
C
C      KMAX = Number of points in eta (marching direction)
C      ETANX > 0 Constant far field distance given by piece-wise linear function
C      = 0 Constant far field distance to march out from initial curve
C      DETA = First grid point spacing in marching direction from initial curve
C      (or 0 for no spacing control)
C      DFEAR = Last grid point spacing in marching direction from initial curve
C      (or 0 for no spacing control)
C
C      If ETANX = 0 then include the following lines
C      NMOD = Number of nodes of piece-wise linear function for far field distance
C      Do N=1,NMOD
C      DHOD(N) = Distance at Nth node
C      Enddo
C      Endif
C
C      SHU = Explicit smoothing coefficient 0(1)
C      TIM = TIM factor for smoothing in marching direction (0<=TIM<=3)
C      ITSVOL = Number of times prescribed areas DAREA(J) is smoothed
C
C      JSPER = 0 for non-periodic supplied surface in XI, = 1 for periodic
C      KSPER = 0 for non-periodic supplied surface in eta, = 1 for periodic
C      JSAXSA = 0 if J=1 is not a singular axis point in supplied body surface
C      = 1 if J=1 is a singular axis point in supplied body surface
C      Similarly for JSAXSB for J=JMAX

```

NOV 27 1995 21:45		int.col.i		Page 1	
WING COLLAR SURFACE INPUT FILE					
C	C	File Name :	int.col.i		
C	C	By :	Bret Barton		
C	C	Date :	5 Oct 95		
C	C				
C	C	40, .8, 0.001, 0.0	KMAX, ETANX, DETA, DFAR		
C	C	2	IMARCH		
C	C	-----			
C	C	INPUT NOTES:			
C	C	-----			
C	C	KMAX	=	Number of points in eta (marching direction)	
C	C	ETANX	=	0 March to the end of the underlying grid	
C	C	DETA	=	Distance to march out from initial curve	
C	C	DFAR	=	First grid point spacing in marching direction from initial curve (or 0 to use reference collar grid spacing)	
C	C	IMARCH	=	Last grid point spacing in marching direction from initial curve (or 0 for no spacing control)	
C	C		=	1 March collar in the +coordinate direction	
C	C		=	2 March collar in the -coordinate direction	

```

Nov 27 1995 21:36      collar_hypg.i      Page 1
COLLAR VOLUME GRID - HYPGEN INPUT FILE

File Name : collar_hypg.i
By       : Bret Barton
Date      : 5 Oct 95

The definitions for the variables can be found in
the HYPGEN users manual.

Hints....

Smoothing 'SMU2' should be kept as low as possible
without yielding negative Jacobians or volumes.
Start with a small DZ0() specially in concave or
convex geometries.

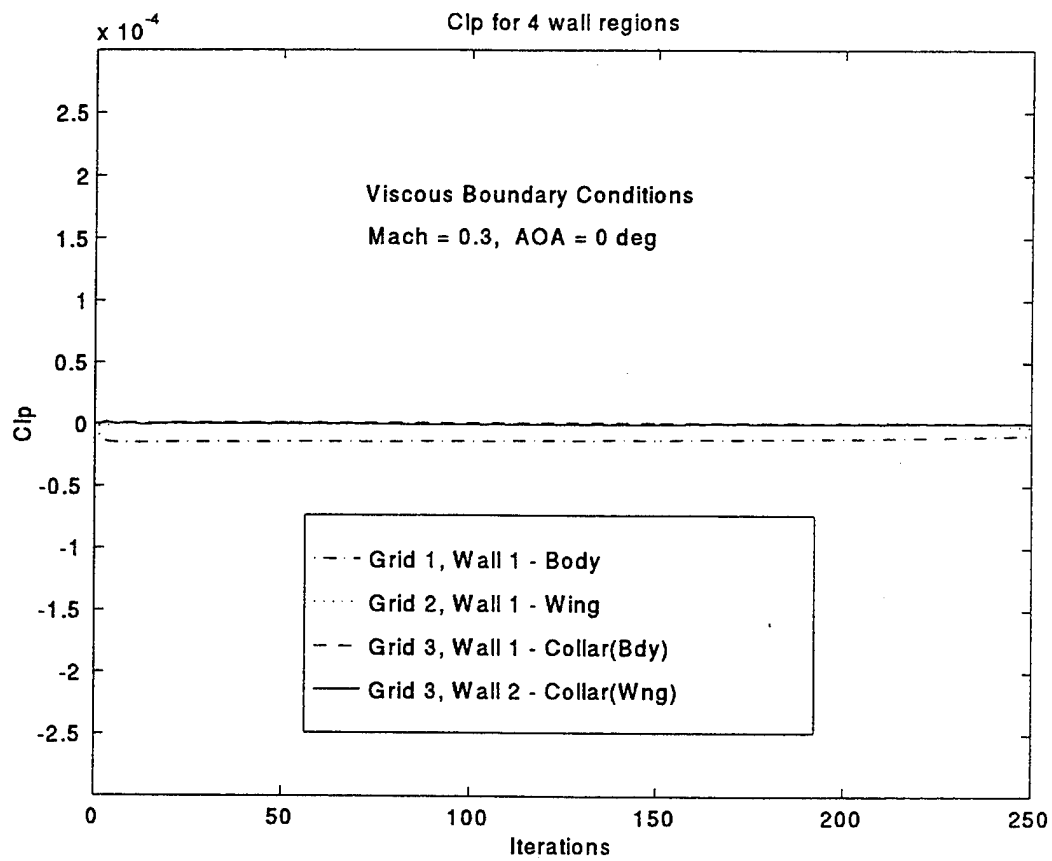
0 2, 1
0 32, .6, 0.00001, 0.0
0 0, 0
0 0, -1, -1, 0, 0
0 0, -1, -1
1, 5.01, 5
0, 0.0, 3
1, 0.001, 0.35

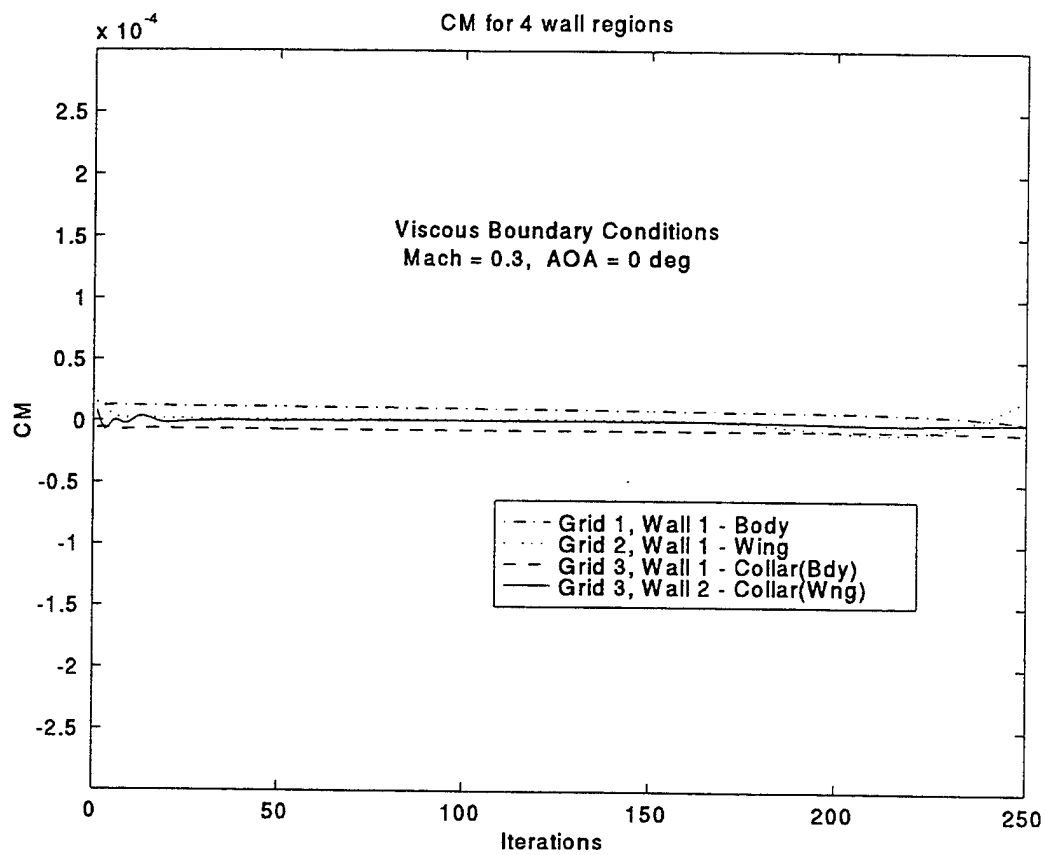
FORM(0/1)
I2SPEC(1/2), N2RES
N2SPEC(1/2), E2SPEC(1/2), DZ1()
JPER(0/1), KPER(0/1)
ISYHJA, ISYHJB, IFLA, IFLTB, IANJA, IANJB
IJSPEC(1/2), EPSSES, ITSVOL
IMETH(0/1/2/3), SMU2
TIMJ, TINK
IAXIS, EXAXIS, VOLUMES

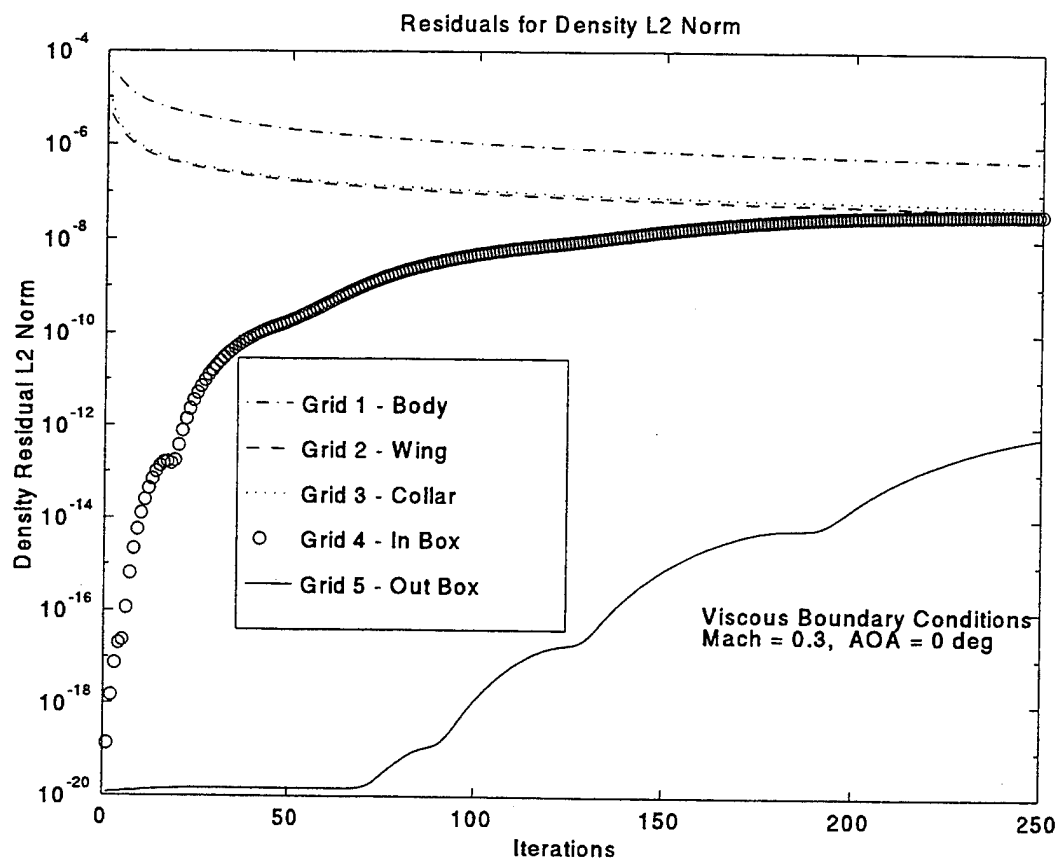
```

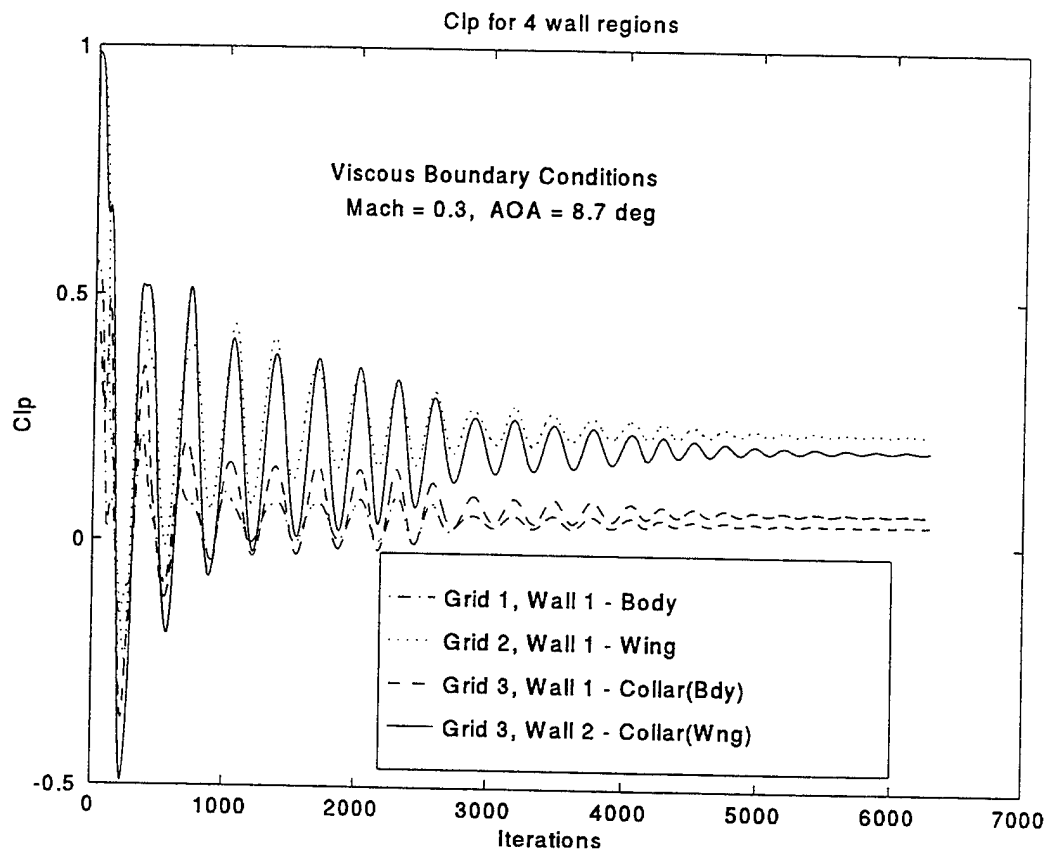
```
C
C      BODY VOLUME GRID - HYGEN INPUT FILE
C
C
C   File Name : body_hyrg.i
C   By       : Bret Barton
C   Date      : 5 Oct 95
C
C
C   The definitions for the variables can be found in
C   the HYGEN users manual.
C
C
C   Hints....
C
C   Smoothing 'SMU2' should be kept as low as possible
C   without yielding negative Jacobian volumes.
C   Start with a small DZ5() specially in concave or
C   convex geometries.
C
C
C   IFORM(0/1), HZREG
C   HZREG(1), ZREG(1), DZ0(1), DZ1(1)
C   JPER(0/1), KPER(0/1)
C   ISYKA, ISYNJB, IFITUA, IFITUJ, IAXSJA, IAXSJB
C   ISYKKB, ISYHKB, IFUTKA, IFUTKB
C   ITPER(1/2), EPSSS, ITNSVOL
C   TMSI(0/2/3), SMO2
C   TIMJ, TINK
C   IAXIS, EXAXIS, VOLRES
C
C   1. 0.0, 0.00
C   1. 0.4, 0.40
C   12. 1.5, 0.00008, 0.0
C   0. 0.0, 0.0
```

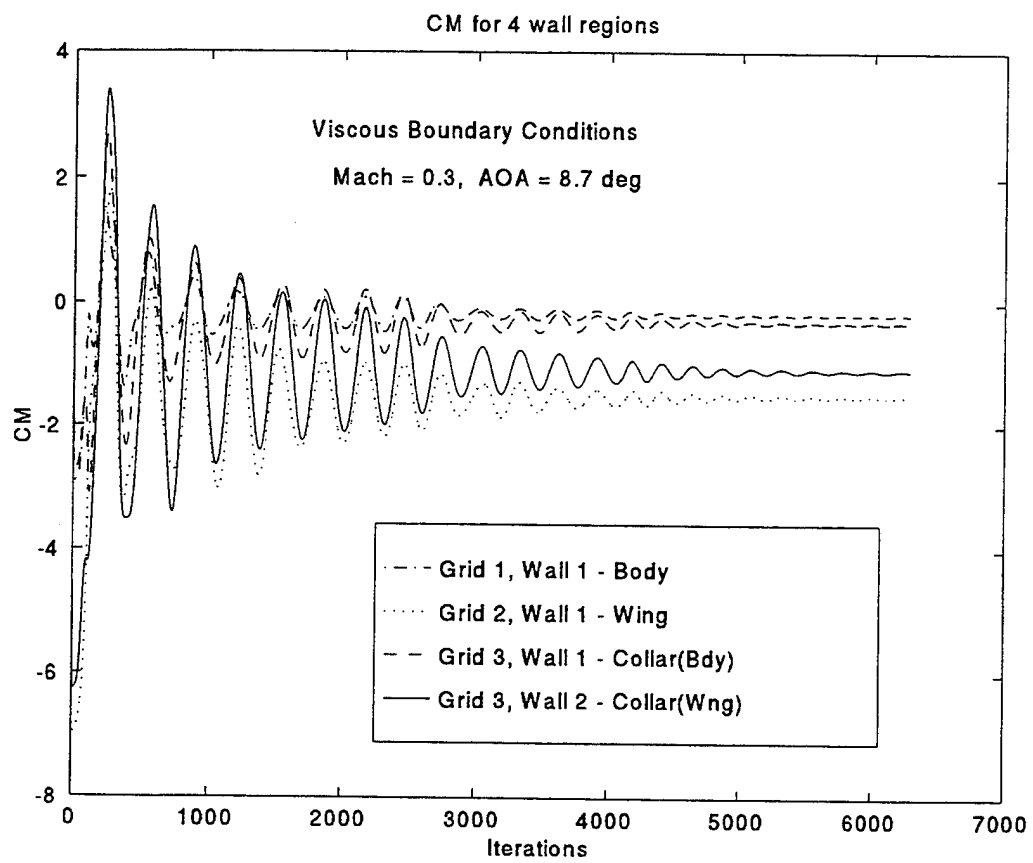
APPENDIX E. L_2 NORM, C_L & C_M PLOTS

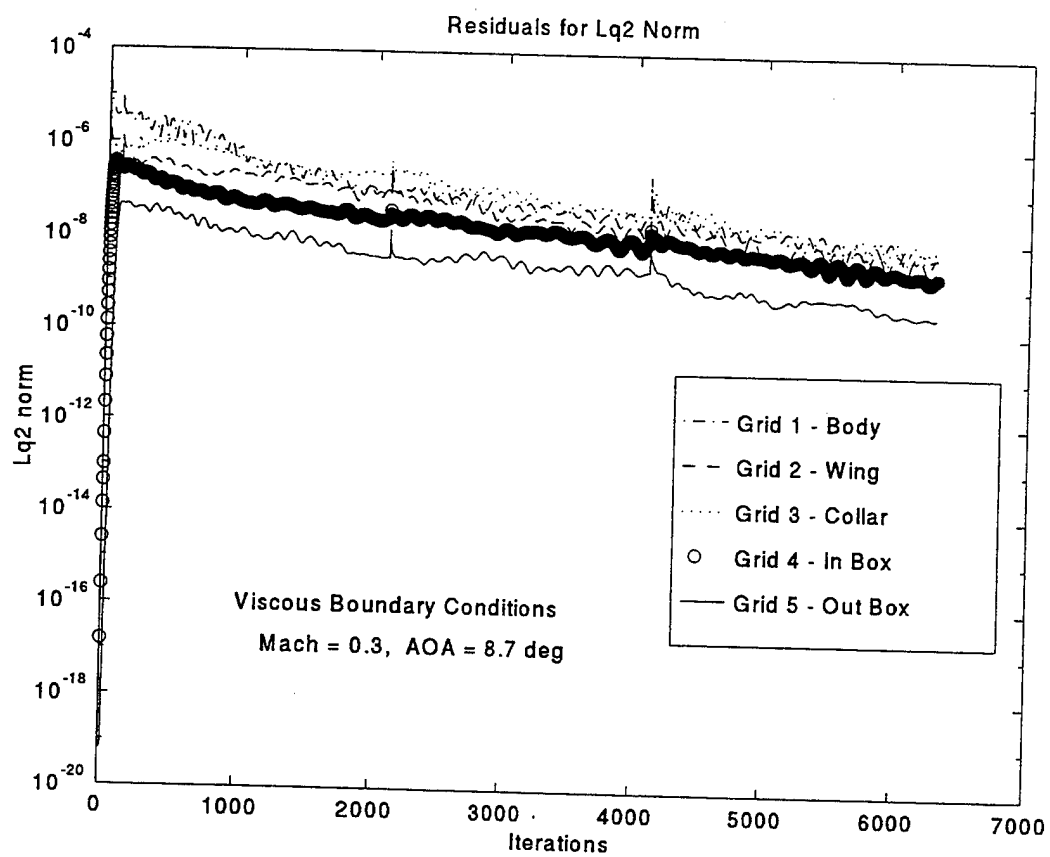


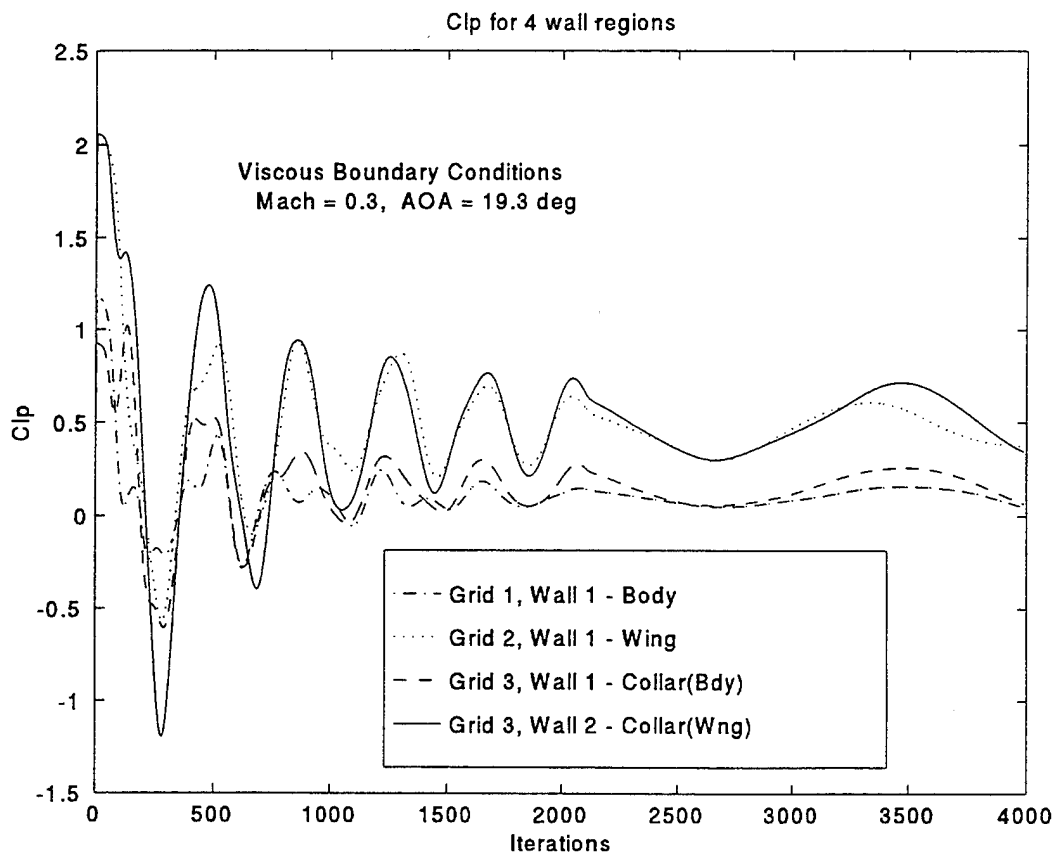


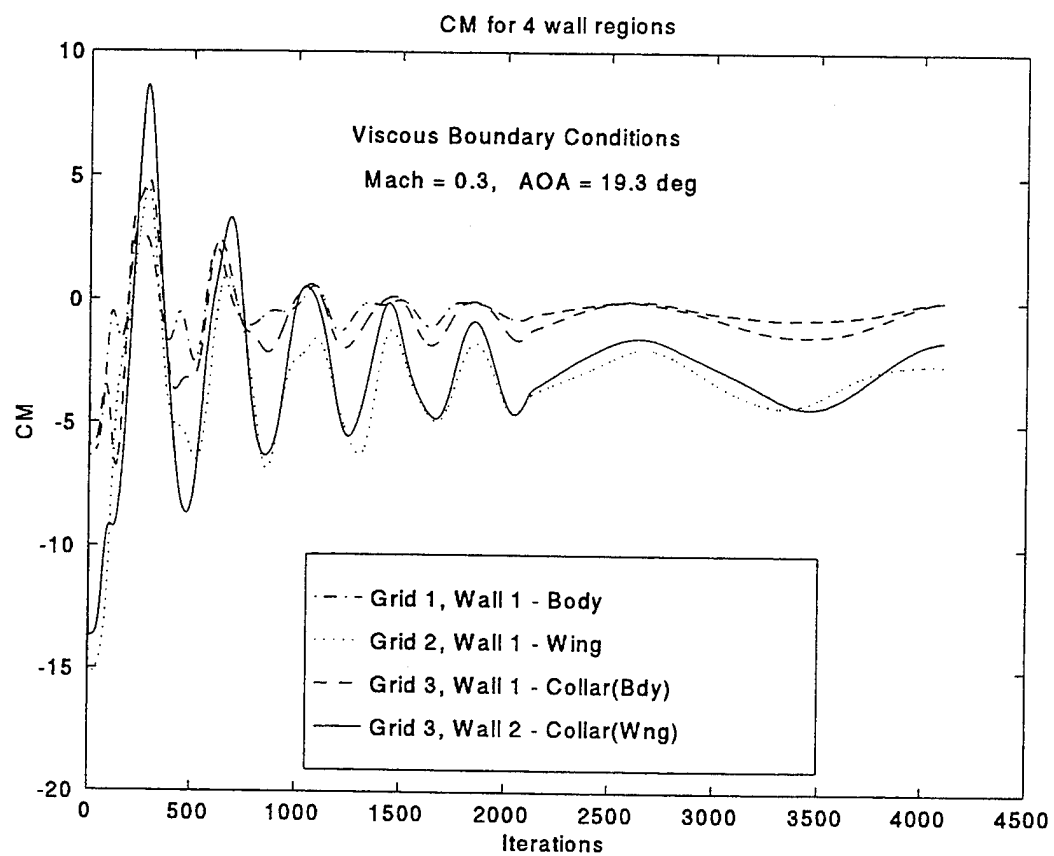


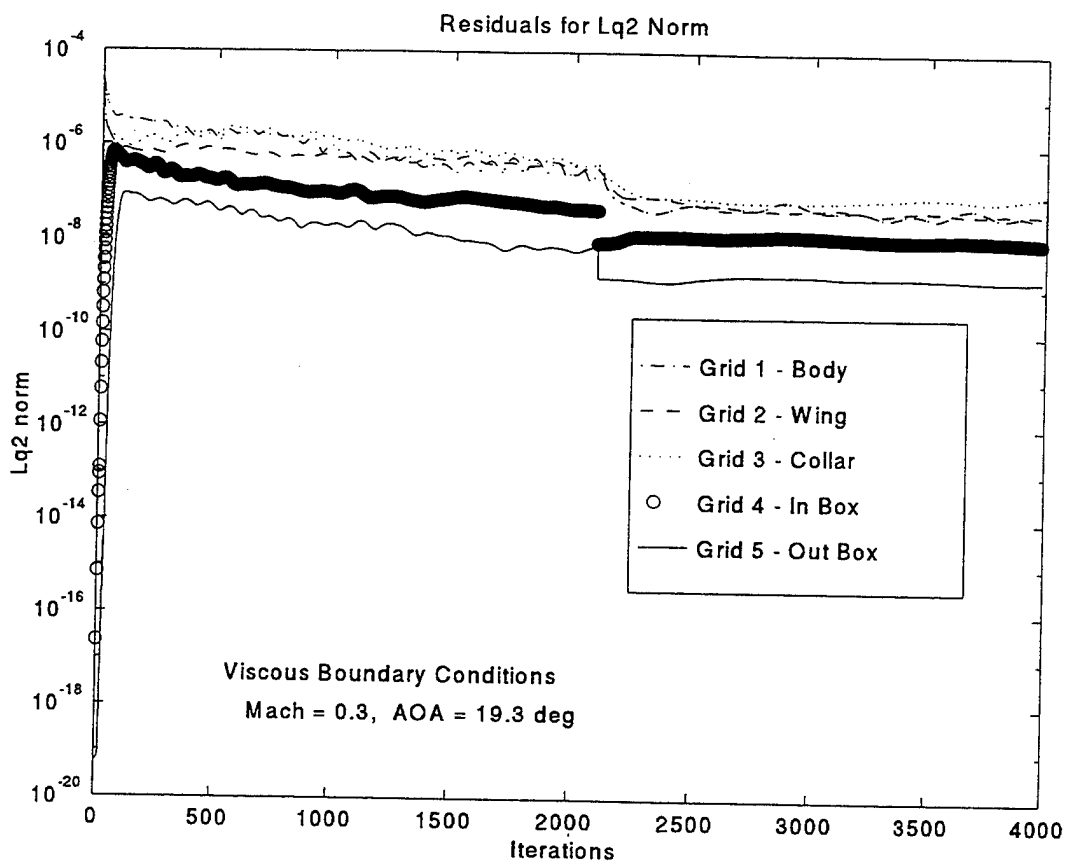












INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center 8725 John J. Kingman Rd., STE 0944 Ft. Belvoir, VA 22060-6218	2
2.	Dudley Knox Library Naval Postgraduate School Monterey, CA 93943-5101	2
3.	Attention: DSIS 3 Director Scientific Information Services National Defence Headquarters Ottawa, Canada, K1A 0K2	1
4.	Naval Air Warfare Center Weapons Division, Code 32463 China Lake, CA 93555-6001	1
5.	Prof. D. J. Collins, Chairman, Code AA/Co Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943-5106	1
6.	Prof. G. Hobson, Code AA/Ag Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943-5106	5
7.	Prof. O. Biblarz, Code AA/Bi Department of Aeronautics and Astronautics Naval Postgraduate School Monterey, CA 93943-5106	1
8.	Ron Marvin 818 Kevin Way Ridgecrest, CA 93555	1

- | | | |
|-----|---|---|
| 9. | Peter Buning
c/o Prof. G. Hobson, Code AA/Ag
Department of Aeronautics and Astronautics
Naval Postgraduate School
Monterey, CA 93943-5106 | 1 |
| 10. | Capt. B. Barton
P.O. Box 2337
Medley, Alberta
Canada, T0A 2M0 | 2 |